

# **Розробка мобільних додатків засобами блочного програмування**

Навчально-методичний посібник

УДК

Власій О.О., Винничук М.Д. Розробка мобільних додатків засобами блочного програмування: Навчально-методичний посібник.

Прикарпатський національний університет імені Василя Стефаника, 2021р. 130 с.

Рецензенти:

Шарин С.В. доктор фізико-математичних наук, професор, проректор з науково-педагогічної роботи Прикарпатського національного університету імені Василя Стефаника

Кохановська О.В. - доктор педагогічних наук, доцент, професор кафедри теорії й методики викладання навчальних дисциплін закладу «Херсонська академія неперервної освіти» Херсонської обласної ради

В навчально-методичному посібнику розглянуто можливості середовища візуальної розробки Android додатків MIT App Inventor 2, вивчення якого можна розглядати як логічне продовження вивчення блочного програмування в Scratch. В посібнику охарактеризовано специфіку роботи в середовищі MIT App Inventor 2, наведено огляд можливостей цього середовища, наведено програму навчального курсу з розробки Android додатків засобами блочного програмування в MIT App Inventor 2, подано дидактичні матеріали для проведення занять.

Посібник буде корисний школярам, студентам, вчителям та викладачам, які цікавляться блочним програмуванням.

© Власій О.О., Винничук М.Д.,  
2021

## ЗМІСТ

Вступ.....	4
ЧАСТИНА 1 ТЕОРЕТИЧНІ ОСНОВИ ВИВЧЕННЯ РОЗРОБКИ ANDROID ДОДАТКІВ У СЕРЕДОВИЩІ MIT APP INVENTOR 2 .....	5
Актуальність вивчення блочного програмування.....	6
Загальна характеристика середовища MIT App Inventor 2.....	8
Специфіка роботи в середовищі MIT APP INVENTOR .....	13
ЧАСТИНА 2 МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДЛЯ ВИВЧЕННЯ ПРОЦЕСУ РОЗРОБКИ ANDROID ДОДАТКІВ ЗАСОБАМИ MIT APP INVENTOR 2....	20
Навчальна програма курсу «Вивчаємо MIT App Inventor разом».....	21
Методичні рекомендації до проведення заняття з курсу “Вивчаємо MIT App Inventor разом” .....	29
ЧАСТИНА 3 ІНСТРУКЦІЇ ДЛЯ ПРОВЕДЕННЯ ЗАНЯТЬ.....	35
Познайомимося?.....	36
Відстежуємо рух.....	46
Інтернет на допомогу.....	55
Час помалювати.....	85
Як же ж без математики? .....	76
Моя перша гра .....	87
Довідник у смартфоні .....	99
Інтелектуальні ігри.....	108
Приховані можливості MIT App Inventor .....	118
Використані джерела .....	127

## ВСТУП

В умовах стрімкого розвитку інформаційних технологій, перед вчителями інформатики все частіше постає питання – як заохочувати дітей до вивчення чогось нового на інформатиці, коли ті «виростали зі смартфонами (комп'ютерами)» в руках? Як знайти щось таке, що б їх мотивувало на роботу та давало якісні знання, при цьому, не відхиляючись від навчального плану? Інформатика зараз є однією із найважливіших предметів в школі. За кількістю годин, виділених на її вивчення, вона поступається лише українській мові та математиці. Особливе місце в предметі «Інформатика» посідає вивчення теми «Алгоритми та програми». У навчальних програмах, затверджених Міністерством освіти та науки України для 2-4 класів (2016) [Error! Reference source not found.], 5-9 класів (2017) [Error! Reference source not found.] та 10-11 класів (2018) [Error! Reference source not found.], можна побачити, що вивчення програмування передбачено кожного року починаючи з 3 класу та закінчуючи 11.

Як і будь-що інше, програмування слід вивчати поступово – від простого до складного, переходити одразу до кодування не слід. Саме тому, на початкових етапах вивчення алгоритмізації та програмування рекомендується використовувати середовище програмування Scratch. Але чи доречно повторювати вивчення одного й того ж самого середовища впродовж декількох років? Тим паче як показує практика, бажання дітей створювати цікаві проекти часто випереджає навчальну програму. Таким чином, діти можуть втратити цікавість та мотивацію до вивчення програмування. Тож вчителям необхідно знаходити додаткові шляхи розв'язання цієї проблеми. Одним із таких рішень може стати впровадження в освітній процес MIT App Inventor – середовище візуальної розробки android-додатків, що вимагає від користувача мінімальних знань програмування. App Inventor є розвитком ідей блочного програмування, з яким учні знайомляться в Scratch. В Україні App Inventor ще широко не вивчається, але все більше привертає увагу дослідників.

**ЧАСТИНА 1**  
**ТЕОРЕТИЧНІ ОСНОВИ**  
**ВИВЧЕННЯ РОЗРОБКИ ANDROID**  
**ДОДАТКІВ**  
**У СЕРЕДОВИЩІ МІТ APP INVENTOR 2**



## АКТУАЛЬНІСТЬ ВИВЧЕННЯ БЛОЧНОГО ПРОГРАМУВАННЯ

Навчання дітей шкільного віку програмуванню поширена практика у багатьох країнах світу, але процес створення власного програмного продукту часто насторожує та лякає учнів, адже вони починають усвідомлювати всю складність вивчення тієї чи іншої мови програмування [5].

Кодування передбачає не тільки складання програми, але і врахування різних проблемних моментів, прогнозування наслідків та ще багато інших. Для цього необхідно вміти правильно аналізувати та робити висновки. Чим скоріше дитина почне вивчати програмування, тим швидше у неї сформується логічне мислення [6].

Важливо правильно адаптувати методи та інструменти до різного віку здобувачів освіти. Наприклад, для учнів старших класів цілком прийнятне вивчення мови програмувань високого рівня (Python, C++, JavaScript і т.п.), де для кодування використовується текстовий редактор, але для початкової школи рекомендується використовувати візуальне середовище Scratch. Scratch – середовище та інтерпретована візуальна мова програмування. Програма складається із скриптів, які необхідно збирати із кольорових блоків. Філософія Scratch полягає у заохоченні дітей до вивчення програмування, тому, замість складних команд тут яскраві блоки, з яких складаються скрипти для персонажів – Рудого котика, Собачки, Мишки та інших.

Завдяки привабливому вигляду (див. рис.1), дітям легко засвоїти базові поняття програмування – умови, цикли, змінні, команди, події, масиви, функції тощо, до того ж, Scratch являється ще й об'єктно-орієнтованою програмою, що вчить дітей думати об'єктно.

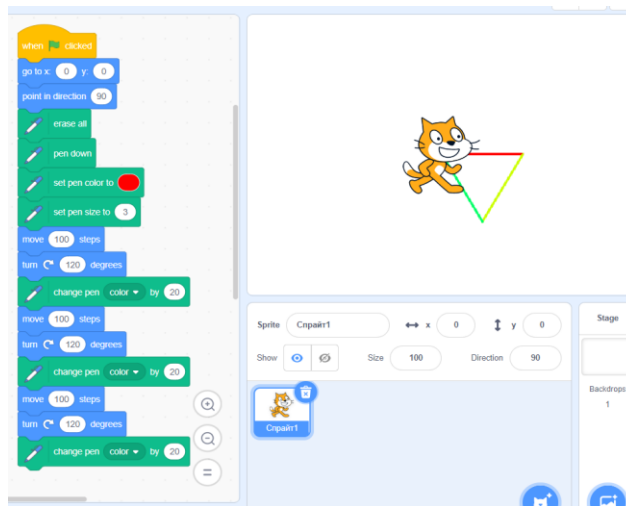


Рисунок 1 Інтерфейс середовища Scratch

Основи програмування розпочинається у 2 класі, де учні вивчаються поняття “команда”, “виконавець”. Продовжується тема у 3 класі поняттям “алгоритм”. На цьому кроці учні вже створюють лінійні програми. Вже у 4 класі учні мають вміти застосовувати у своїх програмах різні види алгоритмів: лінійний, розгалужений, циклічний.

Практично кожен учень має власний мобільний пристрій. За даними Statcounter.com, в березні 2021 року найпопулярнішою операційною системою мобільних телефонів є Android – 71.9% **[Error! Reference source not found.]**. Тому, протягом останніх років, все частіше відбувається інтеграція освітнього процесу з моделлю BYOD (англ. Bring Your Own Device), згідно якої, учні можуть приносити в навчальні заклади свої девайси, щоб використовувати їх в навчальних цілях **[Error! Reference source not found.]**, тож можна запропонувати дітям створювати Android додатки та випробовувати їх на своєму ж смартфоні.

Для подальшого пояснення матеріалу у 5-7 класах також використовується Scratch. Але, починаючи з 8 класу учитель може вже самостійно вибрати мову програмування.

MIT App Inventor за своєю філософією дуже схожий на Scratch. Розробники хотіли максимально спростити процес вивчення програмування. MIT App Inventor має ширший функціонал, ніж його

попередник. Насправді, середовище схоже на айсберг – користувачам видно лише маленьку частинку його можливостей, але під водою знаходиться і недоступна набагато більша частина. Це через головну ідею середовища – мінімальні знання програмування. Для тих, хто хоче більше можливостей, існують доповнення, які можна або завантажити, або ж написати власноруч на високорівневій мові програмування Java. Цей момент дозволить поєднати одразу і вивчення блочного програмування в середовищі App Inventor, і вивчення об'єктно-орієнтованої мови програмування високого рівня Java.

На даний момент, MIT App Inventor практично не вивчається в Україні, хоча, деякі спеціалізовані школи вже звернуло увагу на нього. Так, під час проходження навчально-виробничої практики у Ліцеї імені Романа Шухевича я мала змогу навчати учнів 7 класів програмувати Android додатки.

В закордонних країнах вивчення MIT App Inventor стало досить поширеною практикою, що можна побачити на офіційній сторінці середовища [**Error! Reference source not found.**]. До того ж, на офіційній сторінці можна знайти готові методичні рекомендації та навчальні програми, які можна вільно використовувати у освітньому процесі [**Error! Reference source not found.**].

## **ЗАГАЛЬНА ХАРАКТЕРИСТИКА СЕРЕДОВИЩА MIT APP INVENTOR 2**

Є декілька видів мобільних додатків – веб-додатки, гібридні та нативні [**Error! Reference source not found.**].

Веб-додатки не потребують встановлення на смартфон, адже вони працюють за через веб-браузер. Тобто це, в сутності, звичайні інтернет сайти, які виглядають наче повноцінні додатки, але не мають доступу до інформації на пристрої. Плюсом таких додатків є те, що вони



не займають пам'ять смартфона, але недоліком є те, що для роботи з ними потрібне стійке підключення до інтернету.

Нативні – це додатки, які створюються для певної платформи, наприклад Android, IOS тощо. Вони встановлюються безпосередньо на смартфон користувача. Це додатки, які завантажуються наприклад із Play Market, Apple App Store тощо. Додатки цього типу мають найкращі показники ефективності порівняно із іншими. Створюються вони, найчастіше, за допомогою мови програмування високого рівня, наприклад Python або Java.

Гібридними являються додатки, які є і нативними, і веб додатками. Це додатки, які необхідно встановити на телефон, але в сутності – це веб сайт. Такі додатки створюються за допомогою HTML, JS та CSS.

Але також, є і спрощені рішення для створення мобільних додатків – конструктори. Одним із таких є App Inventor, в якому можна створити нативний додаток.

**App Inventor** – це середовище візуальної розробки Android-застосунків. Завдяки ідеї блочного програмування, для роботи із ним користувачу достатньо мати мінімальні знання з програмування, щоб створити досить складний продукт.

Парадигма програмування в App Inventor дуже схожа на Scratch, але Scratch – виключно навчальне середовище, а додатки, створені в MIT App Inventor можна застосовувати у багатьох сферах, завдяки тому, що на сайті є можливість генерування інсталяційний пакет.

Середовище працює завдяки фреймворку GNU, який реалізує мову Scheme (діалект Ліспа) для платформи Java (Android). Також, для роботи середовища використовували Java-бібліотеку Open Blocks, розроблену в MIT [**Error! Reference source not found.**].

Пробну версію продукту було випущено у липні 2010 року компанією Google. Над його розробкою працювала команда Google Lab (див. рис.2).

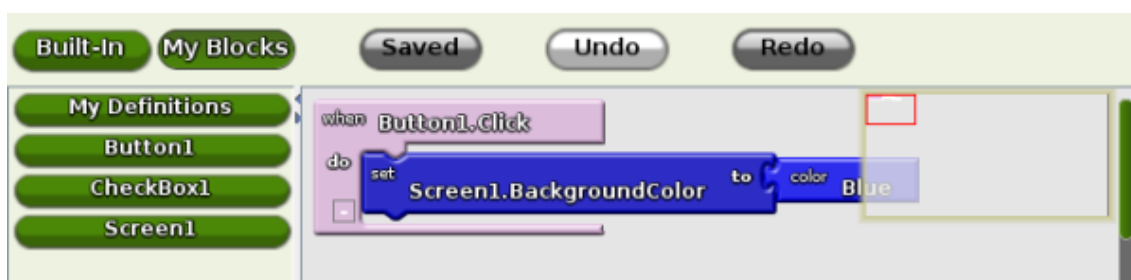


Рисунок 2 Google App Inventor

У 2011 році Google оголосила про закриття лабораторії і подальша робота цієї компанії над проєктом була припинена. Разом з тим, вони хотіли, щоб ідея розвивалася далі, тому вирішили зробити цей застосунок відкритим. Тоді, Массачусетський технологічний інститут, один із провідних технологічних інститутів світу, підхопив ідею та відкрив новий центр мобільного навчання на базі даного програмного продукту, і відповідно, доопрацював середовище. Вже на початку березня 2011 року інститут запустив публічну бета-версію проєкту, доступну на сайті [appinventor.mit.edu](http://appinventor.mit.edu).

Для роботи із середовищем необхідно створити власний обліковий запис, для чого необхідно мати дійсний акаунт Google. MIT App Inventor – хмарне середовище. Розробки та проєктування додатку відбувається в онлайн середовищі, для роботи з яким достатньо лише браузера, що дозволяє працювати із ним за допомогою будь-якого пристрою – Windows, Android, IOS, Linux та інших. Для тестування готового продукту бажано мати мобільний пристрій, або ж можна використовувати емулятор, який є доступним для ОС Windows (див. рис.3 **Error! Reference source not found.**)

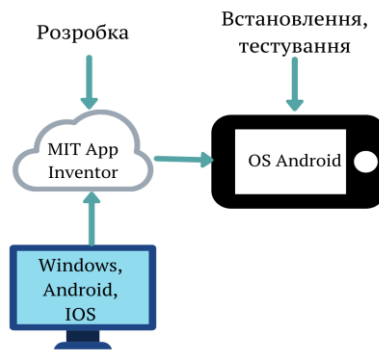


Рисунок 3 Процес проектування та розробки додатку

Після реєстрації, необхідно створити новий проект, перейшовши в меню My projects. Далі, коли вже додаток готовий, його можна зберегти для апробації. Завантажити додаток можна двома способами:

- у вигляді вихідного коду (файл з розширенням .aia). Даний файл надалі можна буде відкрити у середовищі для редагування.
- у вигляді інсталяційного пакета (.apk). В цьому випадку генерується apk файл, який необхідно встановити на Android пристрій.

Для апробації також не обов'язково встановлювати додаток на телефон.

Можна скористатися одним із трьох способів:

- за допомогою програми MIT AI2 Companion (див. рис. 4), яку можна завантажити з Google Play. Тоді, вибравши в меню Connect пункт AI Companion та відсканувавши в додатку згенерований на сайті QR-код, можна переглядати розроблений додаток за допомогою помічника без його безпосереднього встановлення;

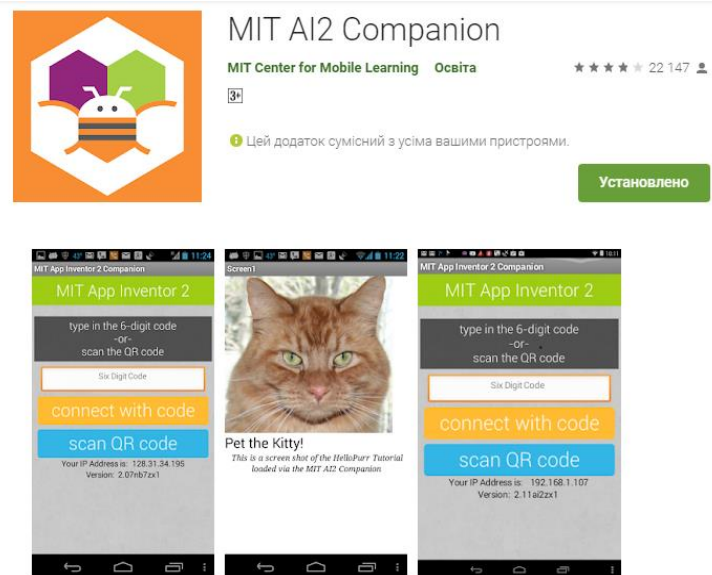


Рисунок 4 MIT AI2 Companion

- за допомогою ПЗ App Inventor Setup Software (див. рис.5), яке потрібно встановити на комп'ютер. Вибравши в меню Connect пункт Emulator, розроблений додаток буде показуватися у встановленому емуляторі;

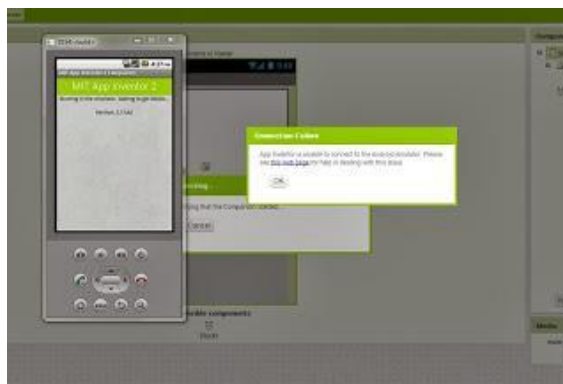


Рисунок 5 App Inventor Setup Software

- за допомогою USB шнура, включивши на телефоні налагодження USB та вибравши в меню Connect пункт USB.

Одними із переваг такого тестування є те, що будь-які найменші зміни у проєкті будуть автоматично показуватися на смартфоні, а також те, що програми не потрібно буде встановлювати безпосередньо на смартфон.

Як і будь-яке інше середовище, AI2 має деякі системні вимоги. Найважливішою є стійке підключення до Wi-Fi, роботи із сервісом відбувається безпосередньо у браузері.

Також, для комфортної роботи з сервісом, версія операційної системи на комп'ютері повинна бути:

- Для Macintosh (з Intel процесором): Mac OS X 10.5 і новіші;
- Для Windows: Windows XP, Windows Vista, Windows 7 та новіші;
- GNU/Linux: Ubuntu 8 та новіші, Debian 5 та новіші.

Найкраще користуватись такими браузерами, як Mozilla Firefox, Apple Safari або Google Chrome **[Error! Reference source not found.]**.

Одним із заохочень для роботи на сайті є конкурс, який проводить команда AI2 - Проєкт місяця **[Error! Reference source not found.]**. Найбільш актуальною темою останніх двох років є пандемія Covid-19, тому, для цієї теми визначено окрему номінацію – MIT APP INVENTOR CORONAVIRUS APP CHALLENGE.

## **СПЕЦИФІКА РОБОТИ В СЕРЕДОВИЩІ MIT APP INVENTOR**

Будь-який додаток складається із двох основних частин. Першою ланкою є інтерфейс, тобто те, що користувач бачить на своєму смартфоні. Це написи, зображення, кнопки або ж інші елементи. Друга ланка це функціонал додатку, тобто кодування. Ці компоненти нерозривно пов'язані між собою. Наприклад, якщо в програму додати кнопку, сама по собі вона нічого не зробить, тобто, її потрібно запрограмувати, або ж навпаки, ми не можемо запрограмувати те, чого у нас на екрані немає.

Саме тому, середовище MIT App Inventor складається із двох режимів - режим проєктування інтерфейсу та режим програмування

елементів програми. Ці процеси реалізуються в окремих вікнах [Error! Reference source not found.].

**Режим «Дизайнер»** – режим в якому створюється інтерфейс додатка. Цей режим використовують для вибору і розміщення різних елементів додатка: кнопок, полів для вводу тексту, написів та ін., які відображаються на екрані мобільного пристрою (див Рис).

**Інтерфейс середовища в режимі редактора можна умовно поділити на чотири частини:**

1. **Палітра** – групи компонентів, які можна додати на екран;
2. **Перегляд** – можливість переглянути один зі створених вами екранів;
3. **Компоненти** – список компонентів, які додані до проекту;
4. **Властивості** – можливість змінювати властивості компонента додатку (наприклад колір, розмір, текст).

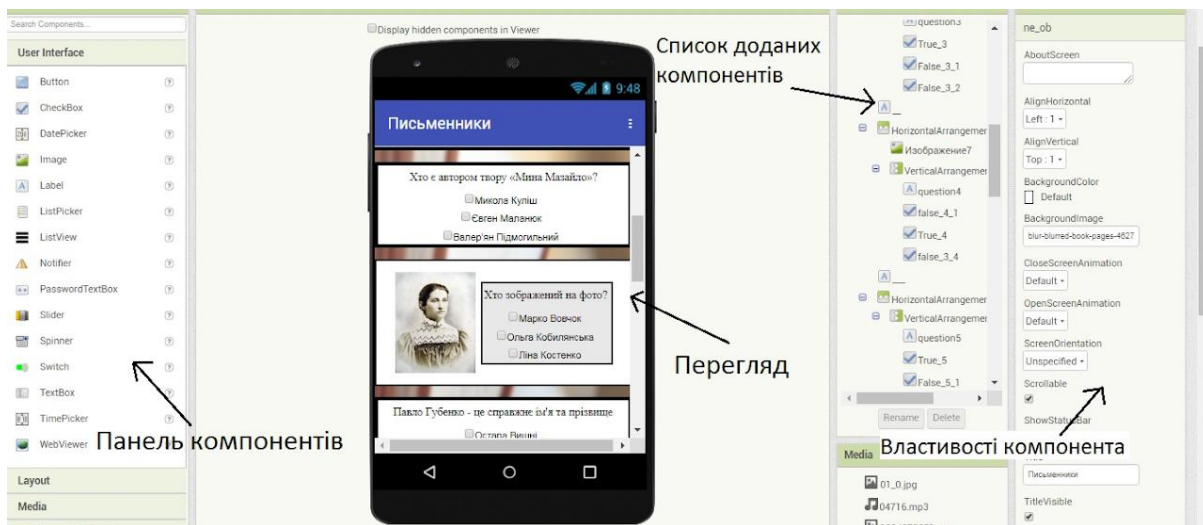


Рисунок 6 Інтерфейс середовища MIT App Inventor в режимі «Дизайнер»

Компонент - це функціональні частини проєкту, наприклад, кнопка, текст, зображення, поля вводу інформації тощо. Деякі компоненти є частиною графічного дизайну, наприклад, кнопки, а деякі - невидимі на екрані пристрою, наприклад, таймер, сенсори або відеоплеєр.

Всі компоненти розділені на 10 груп:

1. *Користувацький інтерфейс* – компоненти для взаємодії з користувачем (Кнопка, Прапорці, Текстове поле тощо).
2. *Розміщення*. За замовчуванням, розміщення компонентів на екрані відбувається у вигляді стовпці, але інколи виникає необхідність розмістити їх у рядок, або у вигляді таблиці. Саме для цього необхідні компоненти із цієї групи – вони дозволяють розмістити інші компоненти "в рядки або стовпці". Тобто, дані компоненти створюють в додатку "таблиці", в комірках яких можна розміщувати інші компоненти.
3. *Медіа* – компоненти, які з'єднують додаток з медіа інструментами, такими як: камера, плеєром, звуком і т.п.
4. *Малювання та анімація*
5. *Карти* – елементи призначені для розміщення карт на екрані смартфона. Основним компонентом з групи є *Map* - елемент середовища, який відображає карти і дозволяє взаємодіяти з ними.
6. *Сенсори* – компоненти цієї групи встановлюють зв'язок дозволяють керувати приладами, вбудованими у смартфон, наприклад, годинник, акселерометр, педометр та інші.
7. *Соціальні мережі* – передбачені для того, щоб здійснити дзвінок, або поділитися деякою інформацією у соціальні мережі.
8. *Сховище* – робота із файлами, збереження інформації незалежно від програми.
9. *Підключення* – Інтернет, Bluetooth тощо.
10. *Lego* – Ці компоненти забезпечують управління роботами LEGO MINDSTORMS NXT за допомогою Bluetooth.

Також, є група з назвою *Експериментальні* (Компонент *Firebase*, який взаємодіє з веб службою для зберігання та отримання інформації).

**Режим «Блоки»** використовується для програмування поведінки додатка і його елементів при різних діях користувача (див Рис).

Інтерфейс середовища в режимі розробника можна умовно поділити на три частини (див. рис. 5):

1. **Групи блоків**, з яких може складатись програма. Натиснувши на бажану групу, відкривається список блоків, які доступні цій групі. Групи блоків можна розділити на загальні, які доступні для всіх компонентів, а також, на блоки, які доступні лише для одного типу компонентів.

2. **Перегляд доступних для вибраної групи блоків**. з яких може складатись програма. Натиснувши на бажану групу, відкривається список блоків, які доступні цій групі. Дана область відкривається після обрання однієї з доступних груп.

3. **Поле** – місце, де можна складати, редагувати та переглядати зібрані конструкції з блоків.

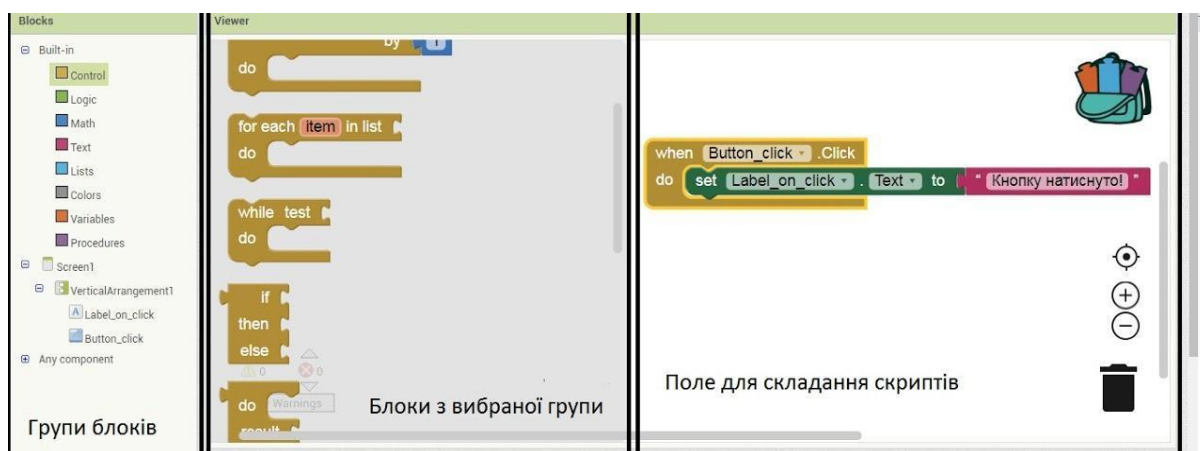


Рисунок 7 Інтерфейс середовища MIT App Inventor в режимі “Розробник”

Блоки в App Inventor - це засоби для програмування поведінки компонентів у вигляді пазлів.

Блоки в цьому конструкторі додатків для Android розбиті на дві великі групи за ознакою того, на що вони впливають і до чого відносяться:

- Пов'язані безпосередньо до компонентів



- Загальні блоки

За допомогою блоків, пов'язаних із компонентом можна змінювати властивості, події чи обробники того чи іншого компонента. Не дивлячись на те, що список елементів програми досить великий, блоки можуть повторюватися та є однотипними для різних компонентів. що полегшує розуміння їх призначення.

Їх можна розділити на три типи:

1. Блоки, які описують властивості компонента. Вони зеленого кольору.

2. Блоки-події, тобто ті блоки, які відстежують настання певних подій. Вони бронзового кольору.

3. Блок-команда або оброблювачі подій. Цей блок вказує, що потрібно зробити з компонентами, до якого належить блок. Вони бузкового кольору.

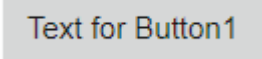
Загальні блоки діляться на 8 підгруп, які, так само як і блоки компонентів, мають свій власний колір.

1. Контролю. Бронзового кольору. В даній групі розташовані блоки, які управляють додатком, а також умовні та циклічні блоки.
2. Логічні. Світло зеленого кольору. Тут розташовані блоки, які використовуються в умовних блоках.
3. Математичні. Вони синього кольору. В даній групі розташовані блоки, необхідні для математичних розрахунків.
4. Текстові. Вони рожевого кольору. В даній групі розташовані блоки, які необхідні для роботи з текстом.
5. Блоки для управління списками. Блоки даної категорії мають блакитне забарвлення. Вони призначені для роботи з масивами.
6. Блоки для управління кольорами. Вони сірого кольору. В даній групі розташовані блоки, які необхідні для задання кольорів. Кольори можна вибрати як зі списку, так і задати їх за допомогою RGB.

7. Змінні. Блоки даної категорії мають помаранчевий колір. Блоки, які належать цій групі ініціалізують глобальні чи локальні змінні, надають їм значення та отримують з них значення.

8. Процедури. Блоки бузкового кольору та використовуються оголошення процедур.

Розглянемо принцип роботи із компонентом Button (Кнопка), який має можливість розпізнавання натискання.

Вигляд кнопки на початковому етапі: . Зовнішній вигляд можна змінити. Це реалізується або в редакторі дизайну, або в редакторі коду за допомогою блоків. Наприклад, можна змінити колір фону, тип шрифту, записаний на кнопці текст, колір тексту, форму тощо,

щоб надати кнопці більш презентабельний вигляд .

Поведінка кнопки регулюється за допомогою обробки подій. Основними для кнопки є:

- Click – натискання кнопки;
- LongClick – довге натискання;
- TouchDown (TouchUp) – кнопка натиснута (відпущена).

Існує задача – після натискання користувачем кнопки, програма має озвучити текст, який зберігається у написі (див Рис). Це подія, отже додаємо блок Click. Далі, вже програмі необхідно дати команду іншому компоненту – Text To Speech, щоб той озвучив текст, тобто, цього разу знадобиться блок-команда бузкового кольору. І останнім етапом необхідно дізнатися, який текст зберігається у компоненті Label, тобто, підглянути в його властивість (зелений колір).

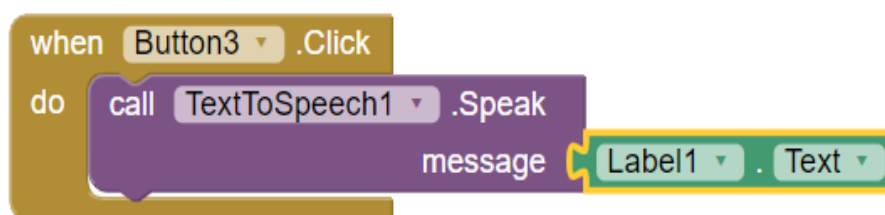


Рисунок 8 Код в середовищі AI2

Різні кольори блоків допомагають швидше засвоїти значення блоків та краще зрозуміти їх призначення, до того складання кольорових скриптів посилює зацікавленість учнів до навчання.

**ЧАСТИНА 2**  
**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ**  
**ДЛЯ ВИВЧЕННЯ**  
**ПРОЦЕСУ РОЗРОБКИ**  
**ANDROID ДОДАТКІВ ЗАСОБАМИ**  
**MIT APP INVENTOR 2**



## НАВЧАЛЬНА ПРОГРАМА КУРСУ «ВИВЧАЄМО MIT APP INVENTOR РАЗОМ»

Метою курсу є формування навичок проєктування та розробки застосунків Android-додатків засобами блочного програмування в середовищі MIT App Inventor 2.

Курс спрямований на формування наступних компетентностей:

*пізнавальної*, яка передбачає опанування основними поняттями програмування, засвоєння технічних та технологічних знань та уявлень з проєктування та створення Android додатків;

*практичної*, яка спрямована на застосування отриманих знань, умінь та навичок програмування, впевненого користування операційною системою комп'ютера та смартфона, онлайн ресурсами, правильного пошуку, зберігання інформації;

*творчої*, яка спрямована на забезпечення розвитку мислення уяви і фантазії; самореалізації у процесі виконання завдань з конструювання; формування інтересу до художньо-технічної творчості, виховання естетичного смаку

*соціальної*, яка спрямована на розвиток трудової культури, досягнення високого рівня освіченості та вихованості; формування кращих особистісних рис (відповідальність, чесність, працелюбство, самостійність, комунікабельність тощо), ціннісного ставлення до себе та інших, вміння працювати в колективі; формування громадянської поведінки, патріотизму, любові до України.

Курс розрахований на учнів 11-14 років.

Головними завданнями курсу є:

- задовольнити індивідуальні освітні інтереси учнів;
- систематизувати знання дітей з курсу програмування;
- ввести для них основні поняття при розробці Android-додатку;

- ознайомити дітей з основними етапами розробки та реалізації мобільних додатків за допомогою MIT App Inventor 2;
- навчити дітей творчо мислити при розробці програм.

Як результат навчання, у дітей повинні сформуватись:

- уявлення про компоненти, блоки, скрипти, процес розробки додатку і т.д.;
- навички розробки програми в середовищі MIT App Inventor 2;
- уміння використовувати базові алгоритмічні структури при розробці додатку;
- алгоритмічне та креативне мислення при розв'язуванні задачі за допомогою програмування.

Викладання курсу передбачає використання технології проєктного навчання. Кожне заняття – це окремий проєкт, розробляючи який, учасники курсу самостійно здобувають необхідні знання та навички для вирішення нових пізнавальних і практичних задач. Такий формат навчання розвиває навички міркування, критичного мислення та синтезу. Знання здобуваються ненав'язливо, природньо, адже для учасників освітнього процесу на першому місці стоїть створення програми, а не просто вивчити матеріал. Коли учасники курсу отримують готовий результат, це мотивує їх розвиватися у даному напрямку.

В основу програми покладені принципи проєктного навчання, від простого до складного, науковості, доступності.

Рекомендована кількість навчальних занять у курсі – 10 навчальних занять. Зміст програми розрахований на 20 навчальних годин.

Навчальна програма складається з двох модулів (по 5 занять).

В першому модулі подаються основні поняття MIT App Inventor 2 та алгоритмічні структури засобами MIT App Inventor. Вивчення цього модуля передбачає розробку одноекранних додатків.

В другому модулі відбувається поглиблене вивчення App Inventor. Діти вивчають створення складніші проекти (ігри, багато екранні додатки). Також, під час вивчення цього модуля учасники курсу персоналізують програми за допомогою власно спроектованого інтерфейсу.

### **Прогнозований результат**

Знати і розуміти:

- правила техніки безпеки під час роботи з персональним комп'ютером;
- поняття коду;
- логічну побудову структур;
- призначення блоків коду.

Уміти і застосовувати:

- використовувати середовище "MIT App Inventor" для створення та виконання додатків;
- обирати структури для написання коду;
- створювати дизайн додатку;
- змінювати властивості компонентів;
- використовувати для оформлення додатків імпортовані елементи;
- використовувати в конструкціях розгалуження та цикли;
- створювати змінні, списки,.

Набути досвіду зі:

- створення мобільних додатків, ігор;
- розробки і презентації власних проектів;
- взаємодопомоги один одному;
- командної роботи.

Структура курсу «Вивчаємо MIT App Inventor разом»

№ заняття	Тема	Зміст заняття	Кількість годин
1	Познайомимося?	Створення облікового запису в середовищі AI2. Ознайомлення з основними компонентами середовища. <i>Перший проєкт «Хто я?»</i>	2
2	Відстежуємо рух	Використання змінних та списків. Випадковий вибір. Робота із сенсором «Акселерометр» <i>Проєкт «Зазирни в майбутнє»</i>	2
3	Інтернет на допомогу	Програмування компонента «Web Viewer». Використання умовних операторів. Логічна умова. <i>Проєкт «Web-переглядач»</i>	2
4	Час помалювати	Особливості компонента Canvas. Координатна сітка. <i>Проєкт «Художник»</i>	2
5	Як же ж без математики?	Створення та робота із власним методом. Додавання параметрів. Виклик методів. <i>Проєкт «Калькулятор»</i>	2
6	Моя перша гра	Створення багатоекранного додатку.	2



		Робота з таймером. <i>Проект «Клікер»</i>	
7	Довідник у смартфоні	Списки. Індексція елементів. Передавання значень між екранами. <i>Проект «Довідник»</i>	2
8	Інтелектуальні ігри	Створення додатку навчального характеру із використанням прапорців, Spinner та Text Box. <i>Проект «Вікторина»</i>	2
9	Приховані можливості MIT App Inventor	Імпортовані компоненти та робота з ними. Програмування розширень <i>Проект «Управління Wi-Fi»</i> <i>Творче завдання: проектування та розробка власного підсумкового проекту</i>	2
10	Підсумкове заняття	Презентація творчих проектів.	2

### Зміст програми

#### 1. Знайомство з MIT App Inventor 2.

*Теоретична частина.* Базові поняття: «додаток», «емулятор», «програма», «компоненти програми», «блок», «кнопки», «напис». Середовище для роботи (сайт <http://ai2.appinventor.mit.edu/>), реєстрація.

Особливості програмування у MIT App Inventor. Інтерфейс середовища. Режими «дизайнер» та «розробка».

Компоненти. Властивості компонентів.

Команди та події в середовищі MIT App Inventor.

Апробація програми. Запуск програми на смартфоні.

*Практична частина.* Програмування кнопок.

Зміна властивостей компонента через код.

Створення програми «Хто я?».

2. Робота зі змінними. Акселерометр.

*Теоретична частина.* Базові поняття: «сенсори», «акселерометр», «змінна», «список».

Випадковий вибір. Принцип роботи генератора випадкових значень.

Інтерфейс власного додатка.

*Практична частина.* Програмування компонента «Акселерометр».

Використання змінних та списків. Різниця між ними.

Створення програми «Заглянь в майбутнє».

3. Компоненти для роботи з Web-сторінками.

*Теоретична частина.* Базові поняття: «браузер», «інтернет-сторінка», «текстове поле», «оператори умови».

Властивості компонента Web Viewer.

Отримання даних від користувача за допомогою компонента «Text Box»

Будова посилань. Інтернет-протоколи та домени.

Призначення логічних блоків та їх різновиди.

*Практична частина.* Програмування компонента «Web Viewer».

Методи компонента.

Використання умовних операторів. Логічна умова.

Створення програми «WEB переглядач».

4. Графіка та засоби малювання.

*Теоретична частина.* Базові поняття: «випадаючий список», «Canvas», «кольори моделі RGB».

Властивості компонента Canvas. Його особливості. Координатна сітка.

Компонент Spinner. Додавання елементів в список.

*Практична частина.* Програмування компонента «Canvas».

Малювання лінії.

Групування компонентів. Зміна властивостей одразу декількох компонентів.

Створення додатку для малювання «Художник».

5. Математичні операції. Проєкт «Калькулятор»

*Теоретична частина.* Базові поняття: «власний метод», «вхідні дані», «математичні блоки».

Призначення та використання власних методів. Метод із параметрами.

*Практична частина.* Створення та робота із власним методом. Додавання параметрів. Виклик методів.

Створення математичного додатка «Калькулятор».

6. Додаток з кількома екранами.

*Теоретична частина.* Базові поняття: «багатоекранний додаток», «таймер», «координати компонента».

Властивості компонента Image Sprite. Його особливості. Координати компонента.

*Практична частина.* Створення багатоекранного додатку. Перехід між екранами. Видимість змінних між екранами.

Робота з таймером. Створення відліку часу.

Створення гри «Клікер».

7. Компоненти для роботи з Web-сторінками.

*Теоретична частина.* Базові поняття: «багатоекранний додаток», «взаємодія екранів».

Списки. Індексція елементів.

Передавання значень між екранами.

*Практична частина.* Створення багатоекранного додатку. Перехід між екранами.

Отримання значення в списку через індекс.

Створення додатку «Довідник України».

8. Робота із прапорцями.

*Теоретична частина.* Базові поняття: «вікторина», «прапорець».

Типи запитань вікторини. Тестування та завдання відкрита відповідь. Методи перевірки відповідей.

*Практична частина.* Створення додатку навчального характеру із використанням прапорців, Spinner та Text Box.

Створення додатку «Вікторина».

9. Розширення MIT App Inventor. Проєкт «Управління Wi-Fi»

*Теоретична частина.* Базові поняття: «розширення», «імпортовані компоненти», «мережа», «конфігурація мережі».

Імпортовані компоненти та робота з ними.

*Практична частина.* Програмування імпортованих компонентів.

Створення додатку для перегляду конфігурації мережі «Управління Wi Fi».

10. Реалізація творчих проєктів. Презентація і захист проєкту.

Розробка проєкту на вільну тему. Розробка і затвердження тем проєктів. Самостійна робота над завданням. Презентація та захист робіт.

## МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ З КУРСУ “ВІВЧАЄМО MIT APP INVENTOR РАЗОМ”

**Тема:** *Познайомимося? Перший проєкт «Хто я?»*

**Мета:** ознайомлення з особливостями та інтерфейсом середовища, формування практичних навичок роботи в різних режимах середовища, формувати творче мислення, отримати навички зі створення проєкту, апробації створеної програми на телефоні.

**Обладнання:** стабільний швидкий інтернет, ПК або ноутбук, Android пристрій.

**Хід заняття**

**Теоретичні основи**

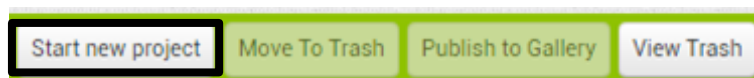
App Inventor – це середовище візуальної розробки android-застосунків, тобто, програмування в ньому відбувається шляхом з'єднання блоків, наче пазлів Lego. Воно знаходиться за посиланням <http://ai2.appinventor.mit.edu/>

Для роботи в ньому необхідно мати власний акаунт, тож

zareestruytsya u n'omu, natishnuvshi knopku



Далі, необхідно створити новий проєкт.



, після чого, введи назву та натисни на кнопку «ОК».

Після попередніх дій відкрився режим дизайнера де ти можеш додавати, змінювати або видаляти компоненти з яких складається твоя програма.

Розробка додатку складається із двох частин – інтерфейс (те, як програма виглядає) та програмування (дії, які програма може виконувати). Так само працює і середовище MIT App Inventor. Спочатку у режимі інтерфейсу потрібно додати всі необхідні компоненти, а тоді перейти до опису їх дій.

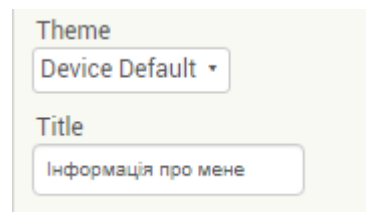
Переміщуватись між ними ти можеш за кнопок Designer та Blocks



у правому верхньому куті.

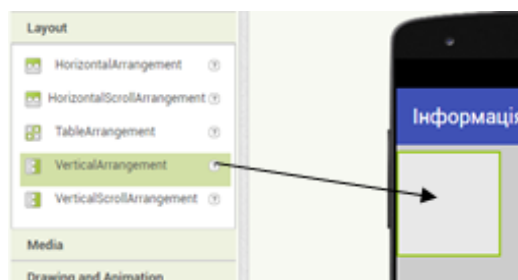
## Частина 1. Розробляємо дизайн

1. Зміни колір фону твоєї програми, вибравши його зі списку доступних для властивості Background



2. В цій же колонці зміни такі властивості Theme та Title так, як це показано на малюнку

3. Перетягни компонент Vertical Arrangement на екран. Він містить лише один стовпець, що допоможе тобі розмістити інші компоненти в рядок.



4. Зміни властивості даного елемента

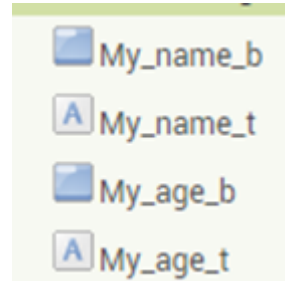
Властивості Align Horizontal та Align Vertical цього елемента відповідають за відцентрування того, що ми будемо мати в середині таблиці.


Властивість Background Color відповідно до назви – це фоновий колір елемента. Зміни його на None щоб зробити таблицю прозорою.

Властивості *Height* та *Width* - це параметри елемента. Зроби Vertical Arrangement рівним параметрам екрану пристрою, на якому запуститься програма.

5. Додай на екран (в середину Vertical Arrangement)

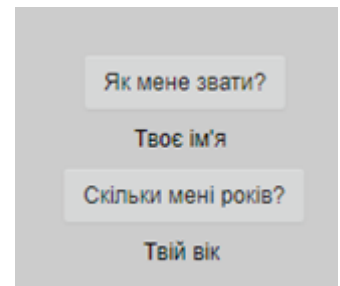
по два елементи  Button та  Label .



6. Перейменуй елементи. Для цього натисни кнопку  внизу переліку доданих компонентів. Введи нову назву та натисни «ОК»

7. Зміни властивість Text для кнопок та написів

Після натиску на елемент відкривається список властивостей для нього. І у кнопки, і у напису є дуже важлива властивість – *Text*. Як ти вже напевне здогадався, змінивши цей пункт ти зміниш текст, який відображається користувачу.



Зміни тексти кнопок та написів на такі (зміни дані відповідно до твоїх).

8. Зроби написи невидимими. Це необхідно для того, щоб інформація відкривалася тільки після того, як користувач натисне на відповідну кнопку і навпаки, ховалась коли вона не потрібна. Для цього потрібно зняти галочку властивості *Visible* для елементів *My\_name\_t* та *My\_age\_t*.

## Частина 2. Програмуємо елементи

9. Перейди в блочний режим

10. Вибери в групі блоків елемент першої кнопки та перетягни на поле такий блок. Даний блок



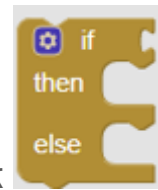
запустить вкладений скрипт після того як користувач натисне на кнопку.


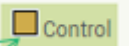
#### 11. Вклади в середину доданого блоку умовний блок «if-else»

Після того, як користувач натисне на першу кнопку, в нього може бути два сценарії розвитку подій:

а) Якщо інформації на екрані немає, тоді потрібно зробити текст видимим, а текст на кнопці змінити на «Сховати ім'я»

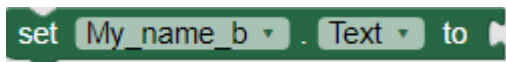
б) Якщо інформація на екрані вже є, тоді потрібно її сховати, а текст на кнопці змінити на «Як мене звати»




Потрібно буде використати умовний блок , який ти можеш знайти в групі .

12. Наступним кроком буде заміна тексту на кнопці на «Сховати ім'я» або «Як мене звати?».

13. Для того, щоб змінити текст на кнопці, використай такий блок:



Але, як ти напевне помітив, в цей блок необхідно вставити інший блок, який зміг би містити в собі інформацію текстового типу. Для таких операцій використовуються блоки з групи *Text*. Тобі необхідний такий блок .

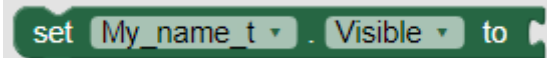
Натиснувши на поле ти можеш змінити текст на свій. Тобі знадобляться два таких блоки, які потрібно вкласти умовний блок.

14. Додай функціонал для написів.



Твій напис повинен з'являтися або зникати в залежності від того чи потрібно показати інформацію чи сховати її.

Для того, щоб робити текст видимим або ні, потрібно додати такий

блок 

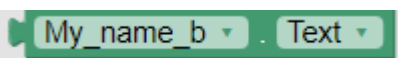

Так як видимість може бути лише або true або false (логічна змінна), попередній блок нам потрібно зв'язати з блоком з групи Logic.

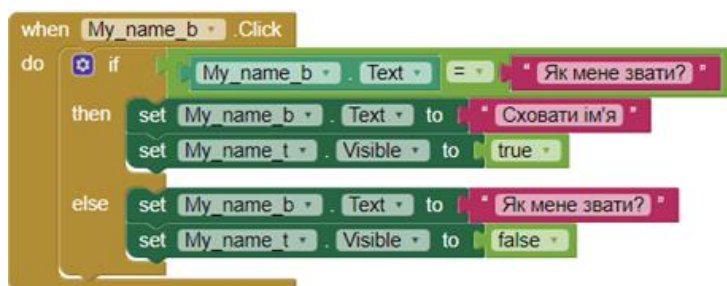
Так само як і в попередньому кроці, тобі потрібні два екземпляри

 та 

15. Додай умову. В умовні блоки потрібно додати відповідно умову. У даному випадку ти будеш перевіряти текст, який написаний на кнопці.

Перевірку ти будеш здійснювати за допомогою логічного порівняння. Цей блок ти можеш знайти тут

Вже знайомі для тебе блоки  та  потрібно вкласти в середину цього блоку так, щоб утворився такий скрипт



```
when My_name_b . Click
do
  if My_name_b . Text = "Як мене звати?"
  then
    set My_name_b . Text to "Сховати ім'я"
    set My_name_t . Visible to true
  else
    set My_name_b . Text to "Як мене звати?"
    set My_name_t . Visible to false
```

16. Повтори попередні кнопки для того, щоб запрограмувати іншу кнопку та напис.

Примітка. Ти можеш скопіювати попередній скрипт та змінити тільки потрібну інформацію.

Змінюй текст на кнопці з «Скільки мені років?» на «Сховати вік» або навпаки та відповідно роби другий видимим або невидимим.

17. Випробуй проєкт. Для цього, завантаж на смартфон додаток MIT AI2 Companion з Play Market. Відкрий його, та натисни обери “Scan QR code”.

Відкрий меню Connect в середовищі MIT APP Inventor на комп’ютері

та вибери зі списку AI Companion.

The image shows a screenshot of the MIT APP Inventor web interface. At the top, there are three menu items: 'Connect', 'Build', and 'Settings'. The 'Connect' menu is open, and a dropdown list is visible below it. The first item in the list is 'AI Companion', which is highlighted in a bright green color. The text 'AI Companion' is also visible in the text area of the page below the screenshot.

Відскануй за допомогою смартфона код, та зачекай на завантаження додатку.

### **Завдання для самостійного виконання**

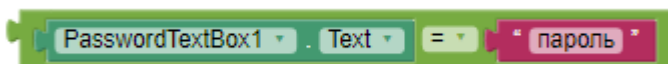
1. Додай інформацію про твоє місце народження (аналогічно до попередніх, у тебе повинна бути ще одна кнопка та ще один надпис, який буде зникати або з’являтися на екрані).

2. Зміни властивості екрану, кнопок та написів на вподобані тобою (колір, шрифт тощо).

3\*. Використовуючи компонент Password Text Box зроби так, щоб інформація з’являлась тільки після правильного вводу пароля.

Примітка. Тобі знадобиться ще один

умовний блок з такою умовою:



**ЧАСТИНА 3**  
**ІНСТРУКЦІЇ**  
**ДЛЯ ПРОВЕДЕННЯ ЗАНЯТЬ**



Що буде в цьому проєкті?

Ти створиш свій перший Android додаток, в якому буде міститися інформація про тебе.

Навички, які ти отримаєш

Ти навчишся створювати проєкт, запускати створену програму на телефоні, а також познайомишся із найважливішими компонентами MIT App Inventor

## Умовні позначення



Виконай

Виконуй ці ІНСТРУКЦІЇ одну за одною



Випробуй

Надішли програму на свій телефон за допомогою QR коду



Подумай

Виконай самостійне завдання

## Кольорові підказки

**Назва**  
**компонента**

**Назва**  
**властивості**

**Значення**  
**властивості**



# Частина 1. Розробляємо дизайн

1

Відкрий середовище MIT App Inventor, яке знаходиться за посиланням <http://ai2.appinventor.mit.edu/>

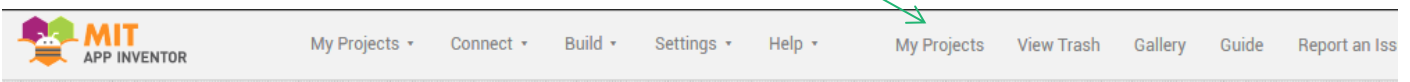
2

Зареєструйся використовуючи свій обліковий запис Google



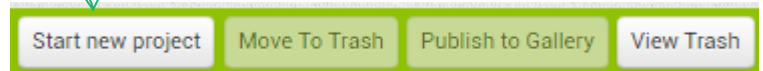
3

Перейди на вкладку «Мої проекти»



4

Створи новий проект



5

Введи назву проекту та натисни кнопку «ОК»

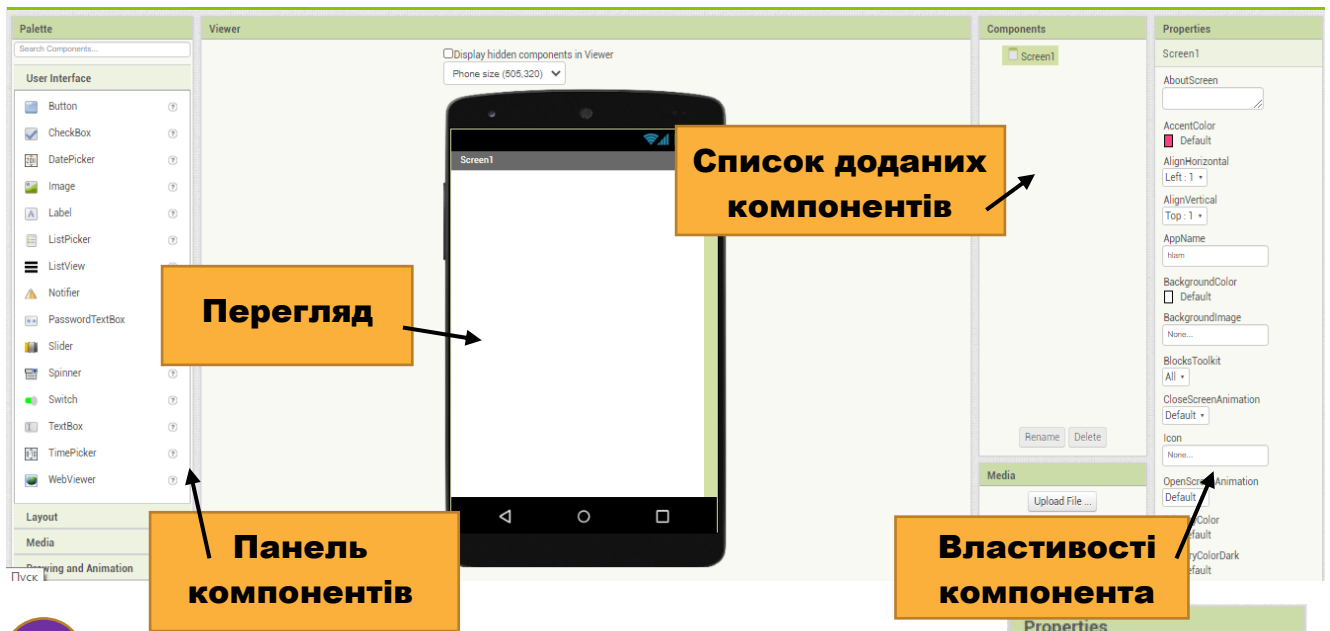
Create new App Inventor project

Project name:

Розробка мобільного додатку в MIT App Inventor відбувається в 2 етапи. Перший етап – проектування інтерфейсу користувача «Як це буде виглядати», другий – програмування елементів програми «Як вони будуть себе вести».

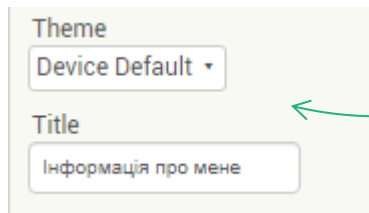


Після попередніх дій тобі відкрився режим дизайнера де ти можеш додавати, змінювати або видаляти компоненти з яких складається твоя програма. Даний режим складається з таких складових:

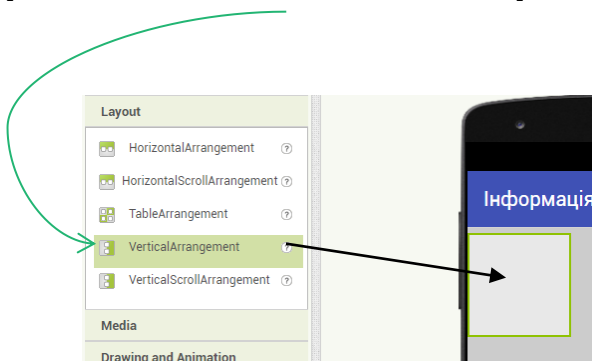


**6** Зміни колір фону (властивість **Background Color**) твоєї програми, вибравши його зі списку

**7** В цій же колонці зміни такі властивості **Theme** та **Title** так, як це показано на малюнку



**8** Перетягни компонент **Vertical Arrangement** на екран. Він містить лише один стовпець, що допоможе нам розмістити інші компоненти в рядочок.



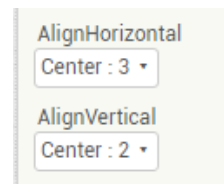
Група компонентів **Layout** допоможе нам гарно розмістити все на нашому екрані, тому ми будемо часто її використовувати. Її елементи - це «таблиці», колонки яких ми заповнюємо іншими компонентами.



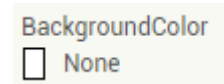
9

**Зміни властивості даного елемента**

Властивості **Align Horizontal** та **Align Vertical** цього елемента відповідають за відцентрування того, що ми будемо мати в середині таблиці.



Властивість **BackgroundColor** відповідно до назви – це фоновий колір елемента. Зміни його на **None** щоб зробити таблицю прозорою.



Властивості **Height** та **Width** - це параметри елемента. Зроби **Vertical Arrangement** рівним параметрам екрану пристрою, на якому запуститься програма.

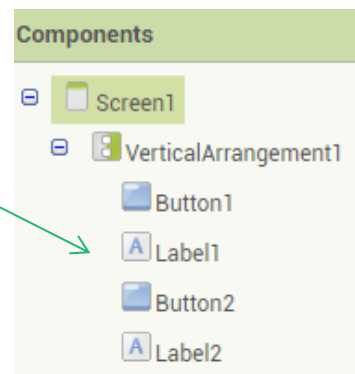


**Група компонентів User Interface містить найважливіші компоненти – кнопку, прапорець, надпис тощо.**



10

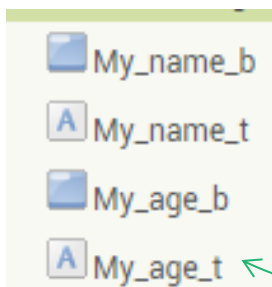
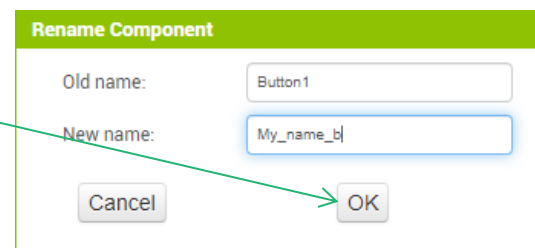
**Додай на екран (в середину **Vertical Arrangement**) по два елементи **Button** та **Label** в такому порядку:**



11

**Перейменуй елементи. Для цього натисни кнопку **Rename** внизу переліку доданих компонентів. Введи нову назву та натисни «OK»**

**Примітка: Називай елементи зрозуміло, так тобі буде простіше пізніше скласти код**



**Дай елементам такі назви**

**12****Зміни властивість `Text` для кнопок та написів**

Text

Як мене звати?

Після натиску на елемент відкривається список властивостей для нього. І у кнопки, і у напису є дуже важлива властивість – `Text`. Як ти вже напевне здогадався, змінивши цей пункт ти зміниш текст, який відображається користувачу.

Зміни тексти кнопок та написів на такі (зміни дані відповідно до твоїх).

Як мене звати?

Твоє ім'я

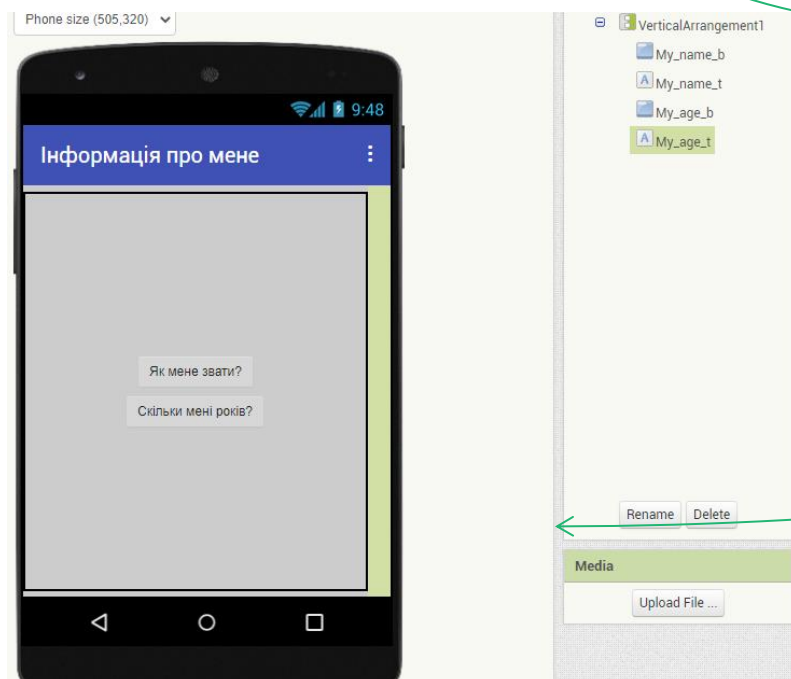
Скільки мені років?

Твій вік

**13**

Зроби написи невидимими. Це необхідно для того, щоб інформація відкривалася тільки після того, як користувач натисне на відповідну кнопку і навпаки, ховалась коли вона не потрібна. Для цього потрібно зняти галочку властивості `Visible` для елементів `My_name_t` та `My_age_t`.

Visible

**14**

Після виконання всіх попередніх кроків, в тебе мав вийти приблизно такий результат





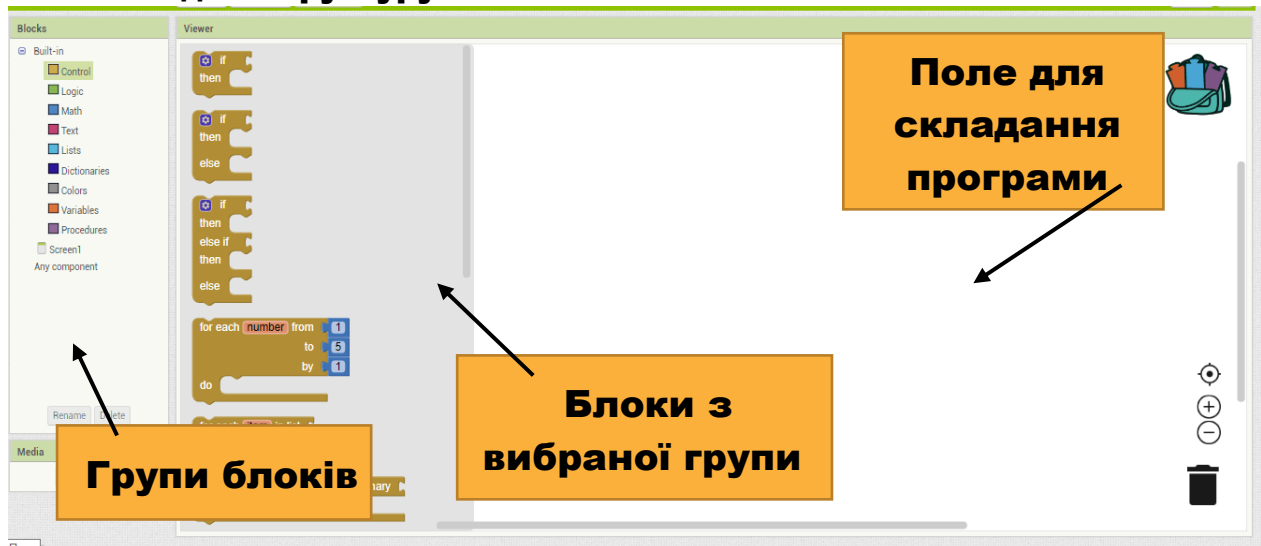
## Частина 2. Програмуємо елементи

15

Перейди в блочний режим.



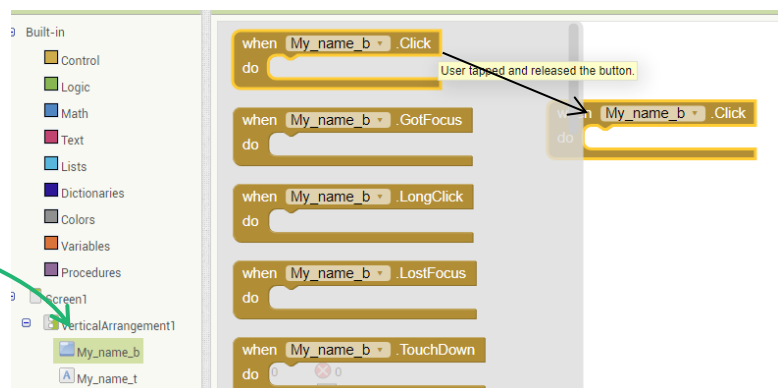
Блочний режим або, як його ще називають - режим розробника, має такий вигляд та структуру.



16

Вибери в групі блоків елемент першої кнопки та перетягни на поле такий блок.

Даний блок запустить вкладений скрипт після того як користувач натисне на кнопку.



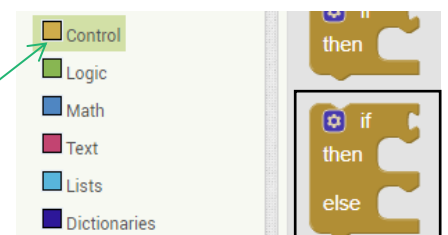
17

Вклади в середину доданого блоку умовний блок «if-else»

Після того, як користувач натисне на першу кнопку, в нього може бути два сценарії розвитку подій:

1. Якщо інформації на екрані немає, тоді потрібно зробити текст видимим, а текст на кнопці змінити на «Сховати ім'я»
2. Якщо інформація на екрані вже є, тоді потрібно її сховати, а текст на кнопці змінити на «Як мене звати»

Як ти бачиш, потрібно буде використати умовний блок, який ти можеш знайти в групі



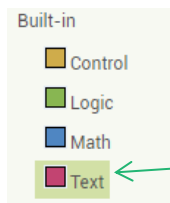
18

Наступним кроком буде заміна тексту на кнопці на «Сховати ім'я» або «Як мене звати?».

Для того, щоб змінити текст на кнопці, використай такий блок:

```
set My_name_b . Text to
```

Але, як ти напевне помітив, в цей блок необхідно вставити інший блок, який зміг би містити в собі інформацію текстового типу. Для таких операцій використовуються блоки з групи **Text**.



Тобі необхідний такий блок



Відповідно, тобі знадобляться два таких екземпляри, які потрібно вкласти умовний блок.

В тебе мав би вийти такий скрипт

```
when My_name_b . Click
do
  if
  then set My_name_b . Text to "Сховати ім'я"
  else set My_name_b . Text to "Як мене звати?"
```

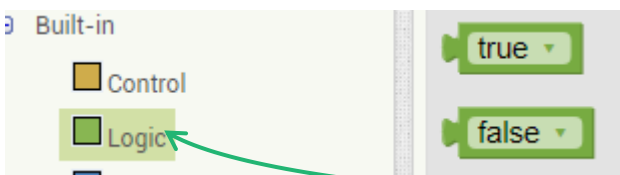
19

Додай функціонал для написів.

Твій напис повинен з'являтися або зникати в залежності від того чи потрібно показати інформацію чи сховати її.

Для того, щоб робити текст видимим або ні, потрібно додати такий блок

```
set My_name_t . Visible to
```



Так як видимість може бути лише або *true* або *false* (логічна змінна), попередній блок нам потрібно зв'язати з блоком з групи **Logic**.

Так само як і в попередньому кроці, тобі потрібні два екземпляри

```
set My_name_t . Visible to true та. set My_name_t . Visible to false
```

Ось такий скрипт виходить

```

when My_name_b .Click
do
  if
  then
    set My_name_b .Text to "Сховати ім'я"
    set My_name_t .Visible to true
  else
    set My_name_b .Text to "Як мене звати?"
    set My_name_t .Visible to false
  
```

20

### Додай умову

Наскільки ти знаєш, в умовні блоки потрібно додати відповідно умову. У даному випадку ти будеш перевіряти текст, який написаний на кнопці.

Перевірку ти будеш здійснювати за допомогою логічного порівняння. Цей блок ти можеш знайти тут



Вже знайомі для тебе блоки  
 My\_name\_b .Text та "Як мене звати?"

потрібно вкласти в середину цього блоку так, щоб утворився такий скрипт

```

when My_name_b .Click
do
  if My_name_b .Text = "Як мене звати?"
  then
    set My_name_b .Text to "Сховати ім'я"
    set My_name_t .Visible to true
  else
    set My_name_b .Text to "Як мене звати?"
    set My_name_t .Visible to false
  
```

21

Повтори попередні кнопки для того, щоб запрограмувати іншу кнопку та напис.

**Примітка.** Ти можеш скопіювати попередній скрипт та змінити тільки потрібну інформацію.

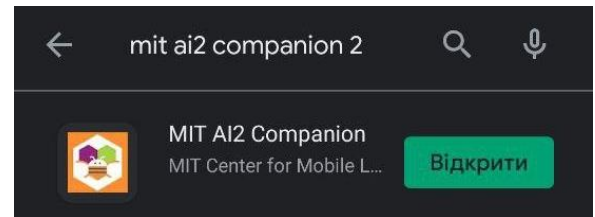
Змінюй текст на кнопці з «Скільки мені років?» на «Сховати вік» або навпаки та відповідно роби другий видимим або невидимим.



## Випробуй свій проект

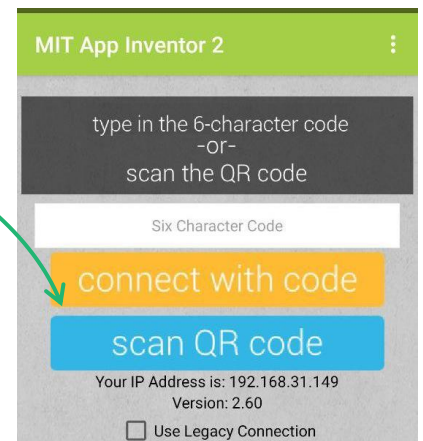
22

Завантаж на смартфон додаток **MIT AI2 Companion** з **Play Market**.



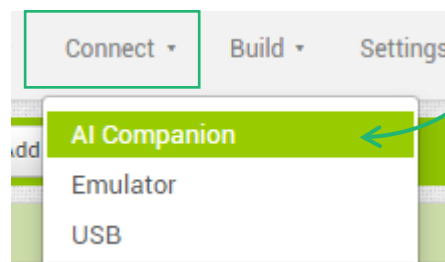
23

Відкрий завантажений додаток **MIT AI2 Companion** та натисни кнопку **Scan QR code**.



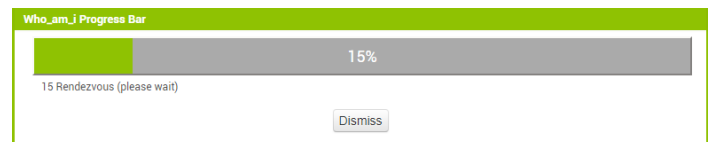
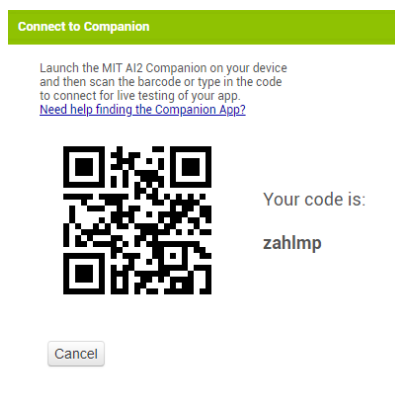
24

Відкрий меню **Connect** в середовищі **MIT APP Inventor** на комп'ютері та вибери зі списку



25

Відскануй смартфоном **QR код**, який відкрився та зачекай завантаження твого додатку



26

Випробуй свій додаток. Спробуй натиснути на одну з кнопок. Чи з'являється необхідна інформація? Чи змінився текст на кнопці? А якщо ти знову натиснеш на цю ж кнопку, інформація зникає?

Спробуй виконати такі ж дії з іншою кнопкою.



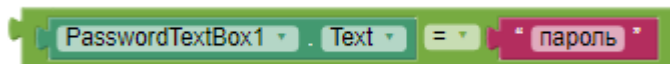
## Самостійні завдання

1. Додай інформацію про твоє місце народження (аналогічно до попередніх, у тебе повинна бути ще одна кнопка та ще один надпис, який буде зникати або з'являтися на екрані).
2. Зміни властивості екрану, кнопок та написів на вподобані тобою (колір, шрифт тощо).

### Спробуй виконати\*

Використовуючи компонент **Password Text Box**  PasswordTextBox зроби так, щоб інформація з'являлась тільки після правильного вводу пароля.

**Примітка.** Тобі знадобиться ще один умовний блок з такою умовою:



Що буде в цьому проєкті?

Гра «Заирни в майбутнє»  
яка дасть відповіді на всі  
твої запитання.

Навички, які ти отримаєш

Ти навчишся працювати із  
рандомом.  
Дізнаєшся що таке  
«змінна» та «список».  
Вивчиш нові компоненти та  
блоки

## Умовні позначення



Виконай

Виконуй ці ІНСТРУКЦІЇ одну  
за одною



Випробуй

Надішли програму на свій  
телефон за допомогою QR  
коду



Подумай

Виконай самостійне завдання

## Кольорові підказки

**Назва**  
**компонента**

**Назва**  
**властивості**

**Значення**  
**властивості**



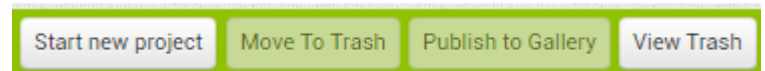
## Частина 1. Розробляємо дизайн

- 1 Завантаж файл *magic\_ball.jpg* на свій комп'ютер. Дане зображення буде фоном програми.

**Примітка №1. Зберігай файли у спеціально створеній для проекту папці.**

- 2 Відкрий середовище MIT App Inventor, яке знаходиться за посиланням <http://ai2.appinventor.mit.edu/>

- 3 Створи новий проєкт

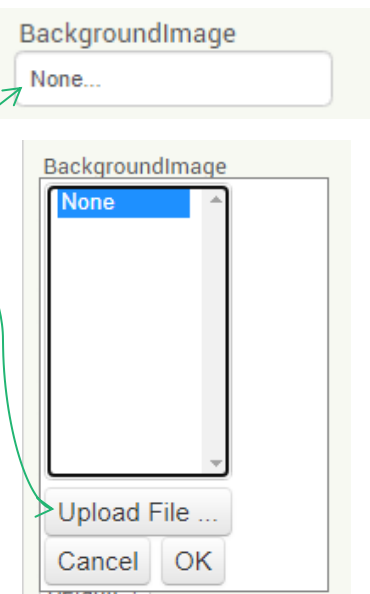
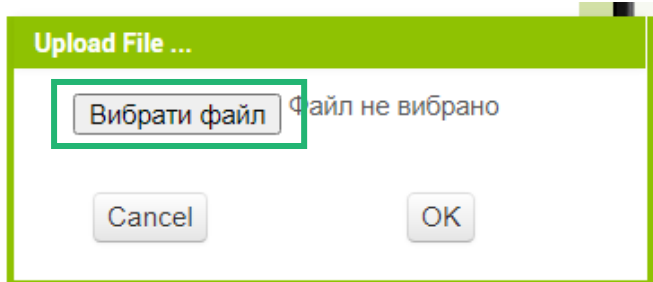


**Примітка. Давай своїм проєктам зрозумілі назви**

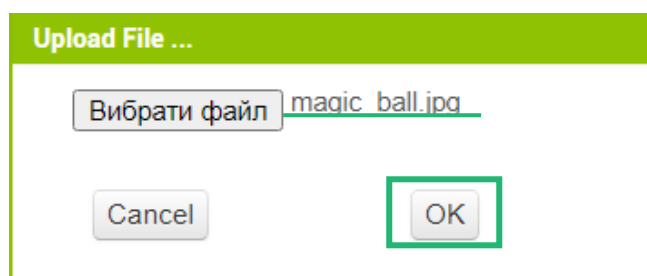
- 4 Встанови фонове зображення екрану. Для цього натисни на значення *None* властивості **Background Image** із списку доступних для компонента **Screen1**.

Натисни на кнопку **Upload File**

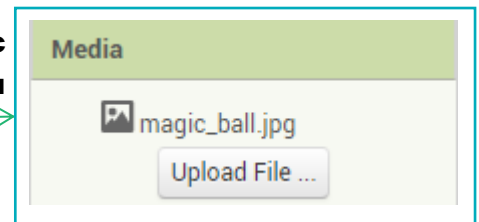
У діалоговому вікні натисни **Вибрати файл**.



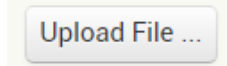
Завантаж *magic\_ball.jpg* з твого комп'ютера та натисни кнопку **OK**.



Тепер завантажено зображення – це ресурс програми. Всі завантажені ресурси ти можеш знайти в переліку **Media**.

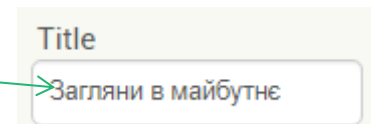


Ти також можеш завантажити файл безпосередньо в цей список, не прив'язуючи його до будь-якої властивості. Для цього просто натисни на



5

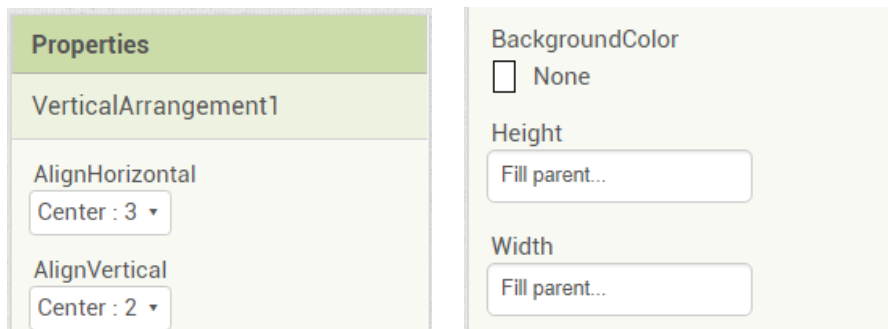
Зміни властивість **Title** на ім'я проєкту.



Також, за бажанням, можеш змінити властивість **Theme**.

6

Додай на екран компонент **Vertical Arrangement** та вибери для нього такі властивості.



7

Додай два елементи напису. Помісти один з них поза межами **Vertical Arrangement**, а інший – всередині нього. Дай елементам зрозумілі назви (наприклад: **Description (опис)** та **Prediction (передбачення)**).

8

Зроби написи читабельними.

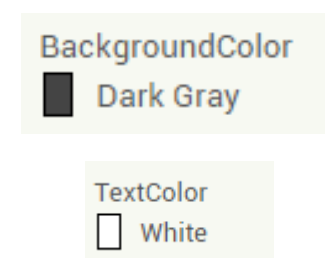
Читабельність є однією з найважливіших ланок правильної побудови інтерфейсу. Хіба ж не так?

**Примітка.** Ти можеш змінювати властивості одразу декількох однотипних елементів. Для цього натисни на **Ctrl** та вибери лівою кнопкою миші необхідні компоненти

Зміни такі властивості написів:

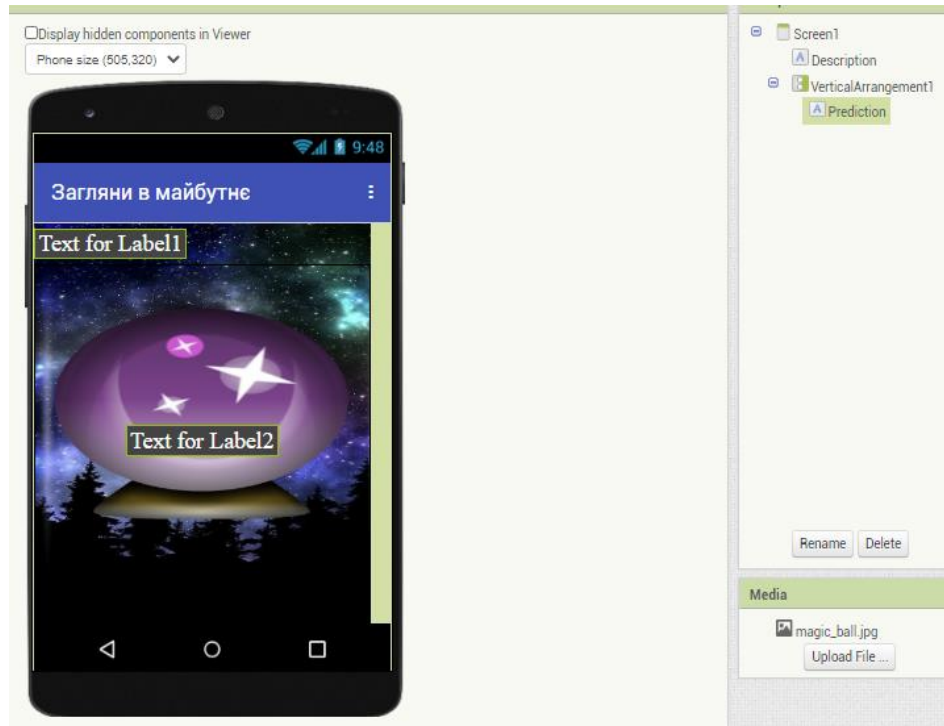
**Background Color** – колір фону тексту. Зміни цю властивість на **Dark Gray**

**Text Color** – колір тексту. Зміни його на **White**

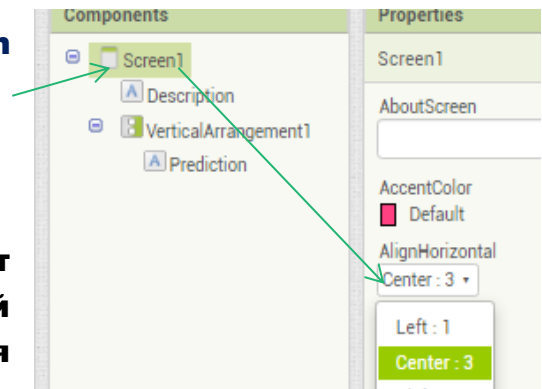




## Результат попередніх дій:

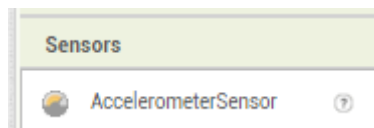


**9** Встанови властивість **Align Horizontal** для екрану на **Center:3**



**10** Додай на екран елемент **Accelerometr Sensor** - компонент, який буде запускати виведення передбачення після струшування телефоном.

Даний компонент належить групі **Sensors**.



**Accelerometr Sensor** – це невидимий компонент, який відслідковує струшування телефону і його переміщення відносно однієї з осей X, Y або Z. За допомогою властивості **Sensitivity** можна задати чутливість акселерометра. (1 – слабка, 2 – помірна, 3 – сильна)



11

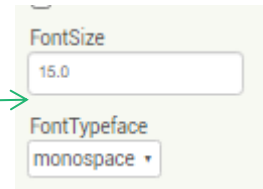
Зміни властивість **Text** для написів.

Перший напис – це підказку користувачу як працювати з програмою. Зміни його наприклад на «*Подумай про своє питання та потруси телефоном*».

Для іншого напису цю властивість залиш порожньою.

12

Налаштуй написи змінивши для них властивості **Font Size** та **Font Typeface**.





## Частина 2. Програмуємо елементи

13

Перейди в режим розробника.



14

Ініціалізуй нову глобальну змінну *List*.

Необхідний блок ти знайдеш в групі 



Завдання цієї змінної: зберігати в собі список можливих варіантів відповідей.

**Змінна** – це математична величина, значення якої може змінюватись у межах певної задачі.



**Список або послідовність** – це

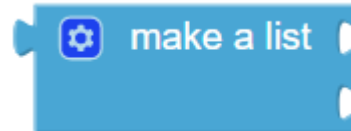
зліченне число впорядкованих значень, де одне і теж саме значення може зустрічатися більше одного разу.



13

Створи список. Для списків, так само як і для змінних, є окремо визначена група

блоків - . Додай на поле такий блок:

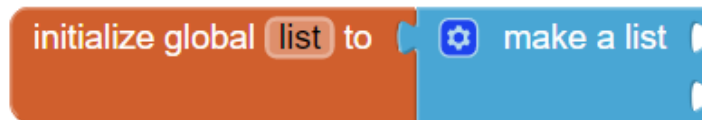
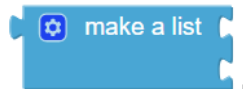


Цей блок – це як частинка пазла. Якщо ти не прикріпиш його до правильного блоку та якщо ти не прикріпиш до нього відповідні блоки – він не спрацює.

З'єднай блоки





та




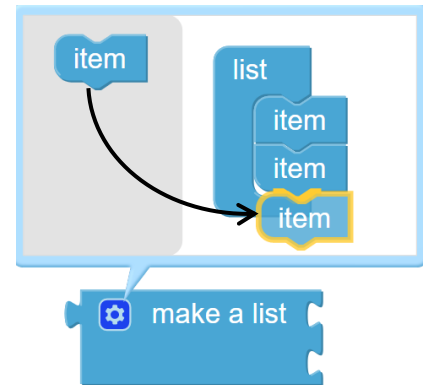
14

Додай до списку перелік можливих відповідей.

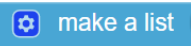
Обери блок  із групи . Зміни текст на «Так» та приєднай його до попередніх блоків.

Додай до списку ще 7 елементів.

**Примітка.** Щоб збільшити кількість елементів у переліку, необхідно відкрити налаштування блоку . Шляхом перетягування додай необхідну кількість елементів.



Ось приблизно таке в тебе мало б вийти:

```
initialize global list to  " Так! "  
" Можливо "  
" Ні "  
" Повторіть питання "  
" Про це не може бути і мови "  
" Не вагайтесь "  
" Успіхів "  
" В наступному житті "
```

15

Запрограмуй компонент **Accelerometer Sensor**.

Додай на поле такий блок. Він запустить скрипт всередині нього після струшування користувачем смартфона.

```
when AccelerometerSensor1 .Shaking  
do
```

16

Спробуй зібрати такий код

```
set Prediction . Text to pick a random item list get global list
```

Змінить текст передбачення

Обере рандомний (випадковий) елемент із списку

Вкаже, з якого саме списку потрібно обирати

17

З'єднай два попередні блоки

```
when AccelerometerSensor1 .Shaking  
do set Prediction . Text to pick a random item list get global list
```



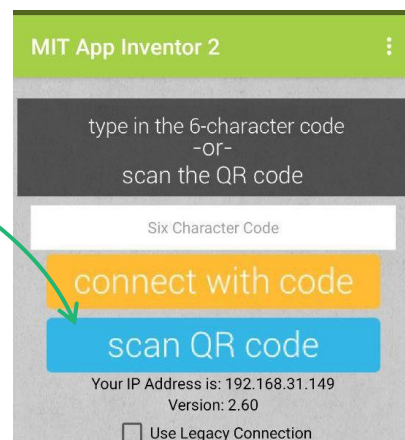
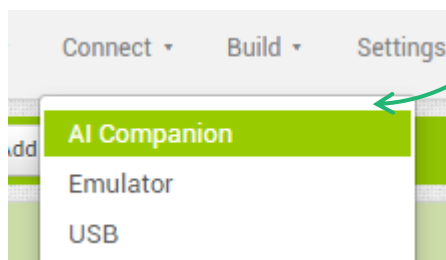
## Випробуй свій проект

18

Відкрий на смартфоні додаток **MIT AI2 Companion** та натисни кнопку **Scan QR code**.

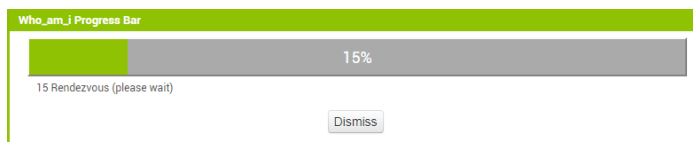
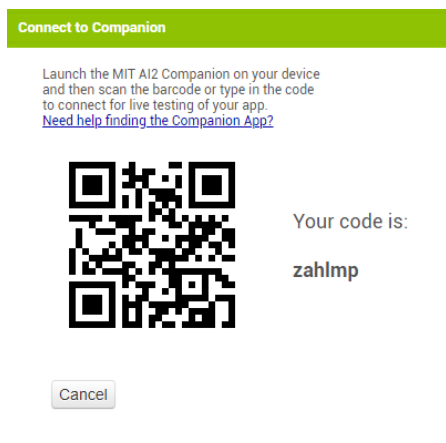
19

Відкрий меню **Connect** в середовищі **App Inventor** та вибери зі списку



20

Відскануй смартфоном QR код, який відкрився та зачекай завантаження твого додатку



21

Випробуй свій додаток.

1. Придумай питання, відповіддю на яке було б «Так» або «Ні»
2. Потруси телефон
3. Прочитай відповідь на своє питання.
4. Поділися з усіма своїм результатом.



## Самостійне завдання

1. Збільш список до 15 елементів.
2. Спробуй змінити властивість **Sensitivity** для **Accelerometr Sensor** на **weak**, **moderate** або **strong** та потруси телефоном. Обери найоптимальніший для твого девайсу режим.

Що буде в цьому проєкті?

Створення Web браузера та  
QR сканера

Навички, які ти отримаєш

Ти навчишся працювати із  
компонентом Web Viewer.

## Умовні позначення



Виконай

Виконуй ці ІНСТРУКЦІЇ одну  
за одною



Випробуй

Надішли програму на свій  
телефон за допомогою QR  
коду



Подумай

Виконай самостійне завдання

## Кольорові підказки

**Назва**  
**компонента**

**Назва**  
**властивості**

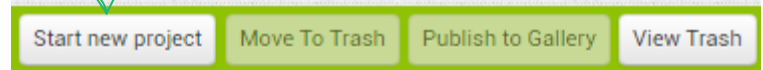
**Значення**  
**властивості**



# Частина 1. Розробляємо дизайн

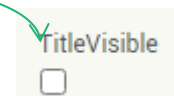
**1** Відкрий середовище MIT App Inventor, яке знаходиться за посиланням <http://ai2.appinventor.mit.edu/>

**2** Створи новий проєкт



**Примітка.** Давай своїм проєктам зрозумілі назви

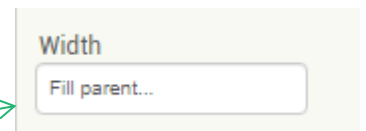
**3** Зроби заголовок програми невидимим



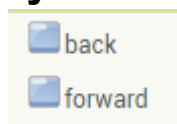
**4** Додай на екран компонент **Horizontal Arrangement**.

В цьому контейнері будуть міститися кнопки для взаємодії з браузером.

Зроби його ширину рівною екрану



**5** Додай в середину дві кнопки – Назад та Вперед. За допомогою них, користувач буде мати змогу повернутися до попередньої сторінки або ж перейти на наступну.

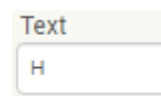


Зміни їх ім'я на **back** та **forward**, а властивість **Text** на **<** та **>** відповідно до їх призначення.



**6** Додай в середину компонента **Horizontal Arrangement** кнопку переходу на домашню сторінку.

Зміни її ім'я на **home**, а властивість **Text** на **Н**

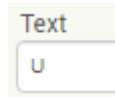


**7** Якої ще важливої функції не вистачає?  
Відповідь – можливість оновлення сторінки.

Створи для цього нову кнопку в середині компонента **Horizontal Arrangement**.



Зміни її ім'я на **update**, а властивість **Text** на

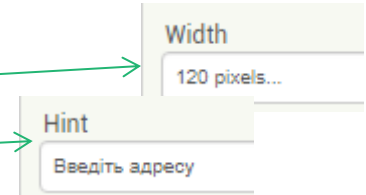


**8** Додай можливість переходу по введеному користувачем посиланню.

За адресний рядок відповідатиме компонент **Text Box**. Додай його в середину **Horizontal Arrangement**.

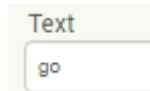
Зміни для нього такі властивості:

- **Ширина** = **120 px**
- **Підказка** = «**Введіть адресу**»




Також, щоб здійснити самий перехід, додай ще одну **кнопку** з ім'ям

**btngo** та властивістю **Text**

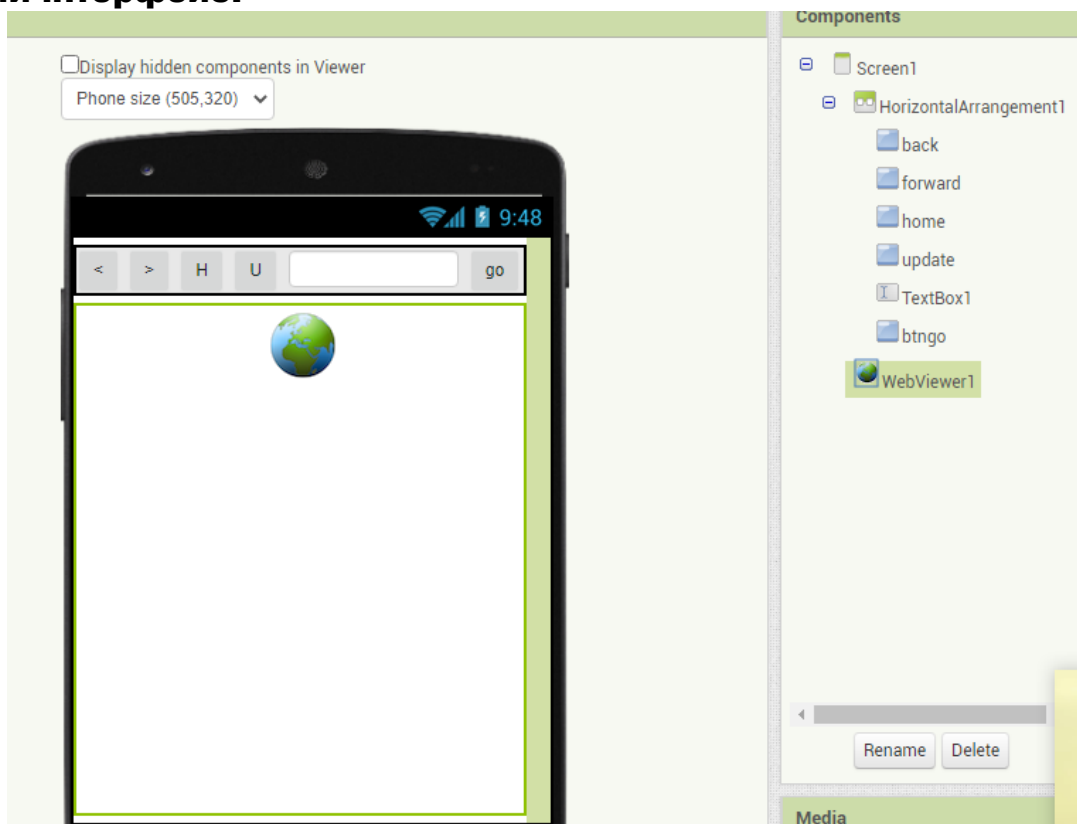


**9** Останнім компонентом стане **Web Viewer** (група **User Interface**)

 **WebView**

Без нього твоя програма не зможе відображати **Web** сторінки.

**Готовий інтерфейс:**





## Частина 2. Програмуємо елементи

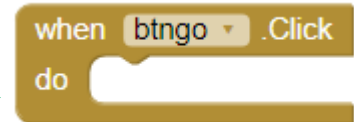
10

Перейди в режим розробника.



11

Спочатку запрограмуй кнопку *btngo* (перехід за посиланням).  
Для цього обери блок



12

Завданням програміста є спрощення життя користувача.  
Спробуй ввести в адресному рядку браузера на комп'ютері посилання на сайт. Наприклад: <http://ai2.appinventor.mit.edu/>  
Що ти бачиш?  
Браузер відкрив дану сторінку.

А зараз, спробуй ввести будь-яке слово.

А так в тебе здійснився пошук по цьому слові в пошуковому сервісі.

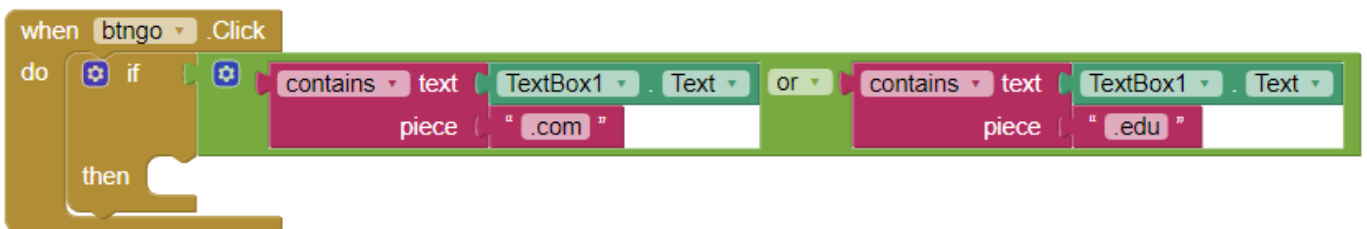
Щоб таке зробити, тобі потрібно використовувати умовні блоки.

Спочатку, зроби перевірку на верхній рівень домену.

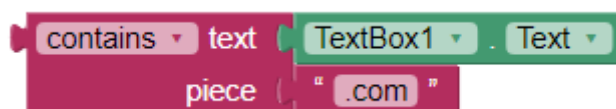
Використаємо для цього оператор *логічного АБО*



**Домен** — це ім'я сайту.  
Він складається з декількох рівнів.  
Перший (верхній) рівень - це частина імені, яка розташована праворуч від останньої крапки.  
Наприклад: .com .edu .ua



Ти можеш додати більше варіантів різних доменів.



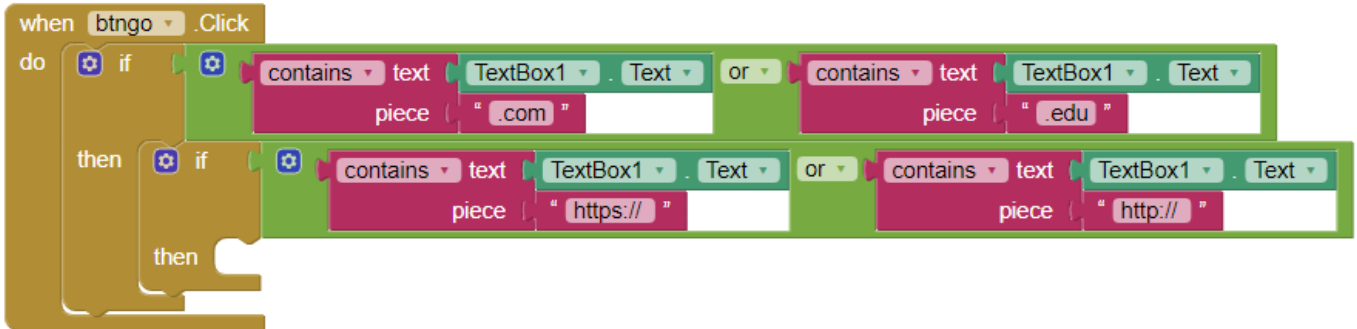
**Примітка.** Блок перевіряє, чи містить текст буквосполучення *.com*

**перевіряє, чи містить**

13

Кожне посилання має початок **https://** або **http://**. Це – інтернет-протокол. Більшість людей не звертає уваги на нього, але без нього браузер не зможе відкрити сайт.

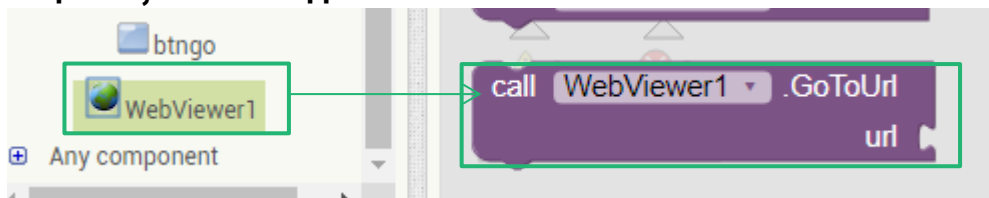
Додай перевірку того, чи прописав користувач протокол чи ні.



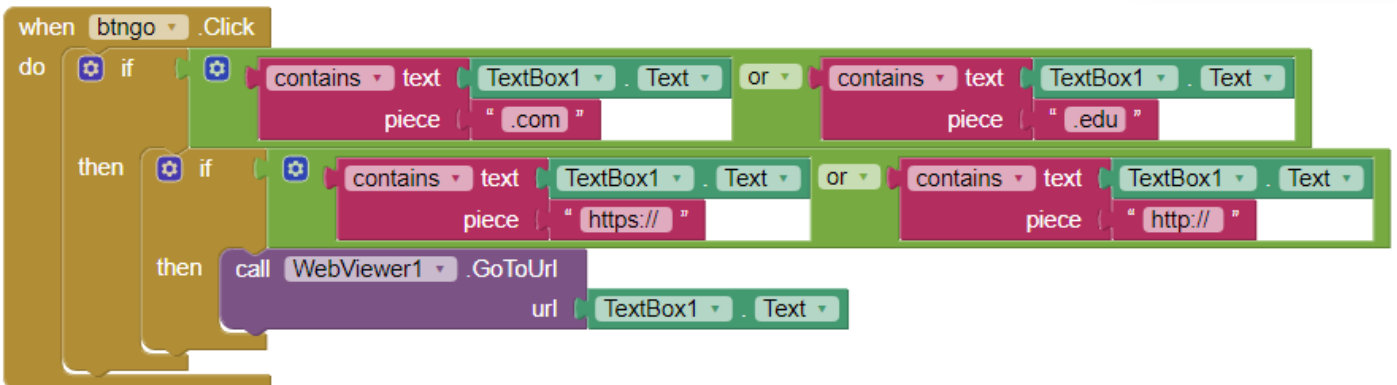
14

Якщо користувач вказав верхній домен та протокол сторінки, тоді здійснюється перехід на вказаний сайт.

Для цього, тобі знадобиться такий блок:



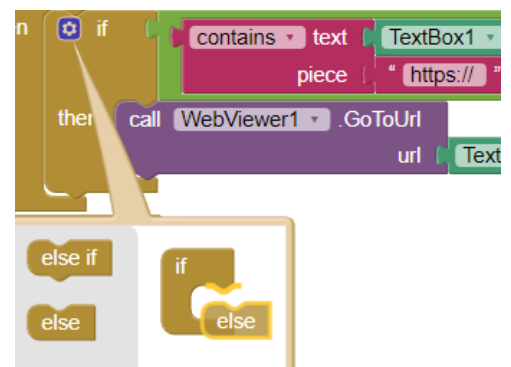
До нього необхідно долучити блок текстового типу.



15

Якщо домен було введено, а протокол сторінки – ні, тобі потрібно просто додати його до посилання.

Тут необхідний оператор **else**

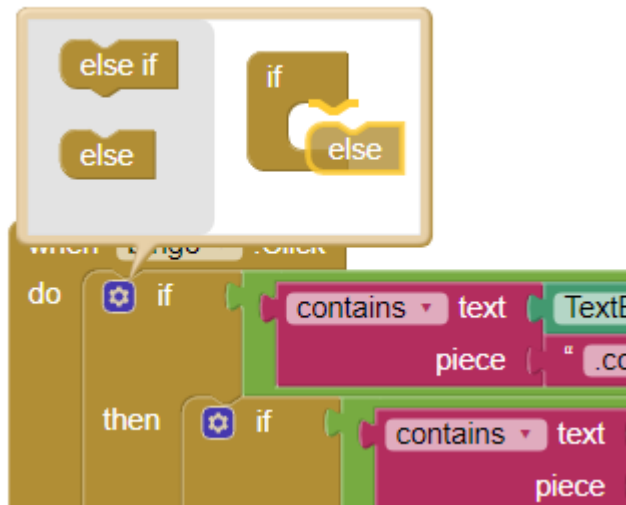


**Додай в цей оператор такий скрипт:**



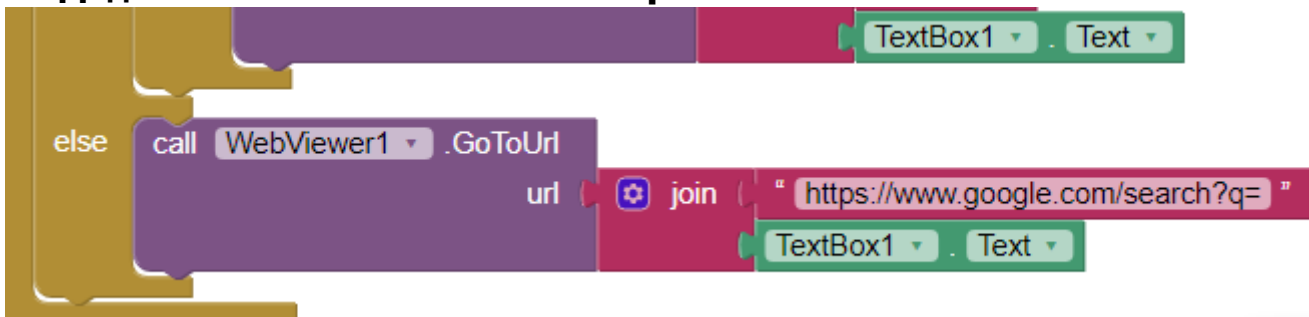
**16** Якщо було введено не посилання – здійснюється перехід на пошуковий сервіс. Нехай це буде Google.

Тобі знадобиться ще один else, тільки цього разу для зовнішнього if



Якщо подивитися на всі посилання в пошуковому сервісі Google, можна побачити аналогію – всі пошукові посилання мають початок <https://www.google.com/search?q=> .

**Додай в зовнішній else такий скрипт**



**17** Під час переходу на різні Web сторінки, в тебе можуть виникнути незначні проблеми. Додай код, щоб їх ігнорувати

```

if [WebView1] . IgnoreSslErrors
then set [WebView1] . IgnoreSslErrors to true

```

**Готовий код для цієї кнопки:**

```

when [btnGo] . Click
do
  if [contains text [TextBox1] . Text or contains text [TextBox1] . Text
  piece ".com" or contains text [TextBox1] . Text or contains text [TextBox1] . Text
  piece ".edu"]
  then
    if [contains text [TextBox1] . Text or contains text [TextBox1] . Text
    piece "https:// " or contains text [TextBox1] . Text or contains text [TextBox1] . Text
    piece "http:// "]
    then call [WebView1] . GoToUrl url [TextBox1] . Text
    else call [WebView1] . GoToUrl url [join "http://" [TextBox1] . Text]
  else call [WebView1] . GoToUrl url [join "https://www.google.com/search?q=" [TextBox1] . Text]
  if [WebView1] . IgnoreSslErrors
  then set [WebView1] . IgnoreSslErrors to true

```

**18**

**Опиши кнопки переходу назад та вперед.  
Для цього є вже готові процедури.**

**Їх виклик здійснюється так:**

```

when [back] . Click
do call [WebView1] . GoBack

when [forward] . Click
do call [WebView1] . GoForward

```

**19**

**Опиши кнопку переходу на домашню сторінку.**

```
when home ▾ .Click
do call WebView1 ▾ .GoHome
```

20

### Опиши кнопку оновлення сторінки

```
when update ▾ .Click
do call WebView1 ▾ .GoToUrl
    url | WebView1 ▾ . CurrentUrl ▾
```



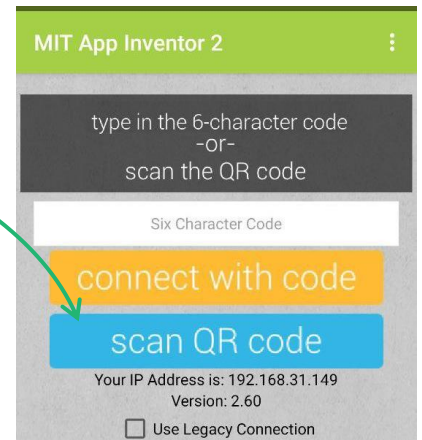
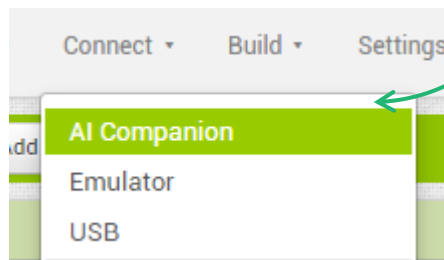
## Випробуй свій проект

21

Відкрий на смартфоні додаток **MIT AI2 Companion** та натисни кнопку **Scan QR code**.

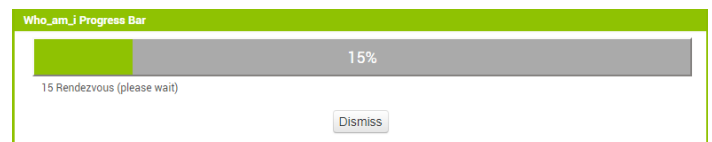
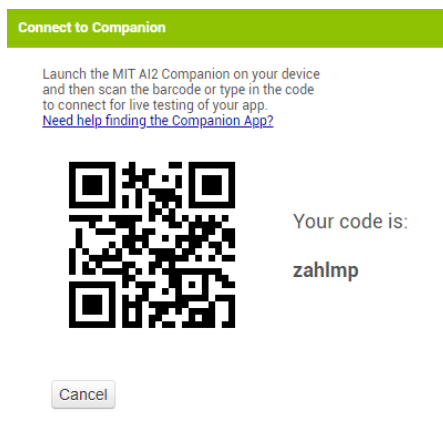
22

Відкрий меню **Connect** в середовищі **App Inventor** та вибери зі списку



23

Відскануй смартфоном QR код, який відкрився та зачекай завантаження твого додатку



24

Випробуй свій додаток.

1. Чи здійснюється перехід за введеним посиланням?
2. Чи працюють всі додані кнопки?
3. Чи відбувається пошук по слову?



## Самостійне завдання

1. Встанови домашню адресу. Ти зможеш зробити це за допомогою властивості **Home Url**

### Самостійне завдання\*

Створи новий проєкт “QR сканер”, який буде переходити по зашифрованому в QR кодi посиланню.

Примітка. Для цього тобі знадобляться такі нові компоненти, як: **Camera** та **BarcodeScanner**.

Код програми:

```
when Button1 .Click
do
  set BarcodeScanner1 . UseExternalScanner to false
  call BarcodeScanner1 .DoScan
  set TextBox1 . Text to BarcodeScanner1 . Result
  call WebViewer1 .GoToUrl
  url BarcodeScanner1 . Result
  set TextBox2 . Text to BarcodeScanner1 . Result
```



Що буде в цьому проєкті?

За допомогою графічних можливостей App Inventor 2 ти створиш графічний редактор.

Навички, які ти отримаєш

Ти познайомишся із такими новими компонентами, як Canvas та Spinner. Вивчиш деякі їх властивості та методи.

## Умовні позначення



Виконай

Виконуй ці ІНСТРУКЦІЇ одну за одною



Випробуй

Надішли програму на свій телефон за допомогою QR коду



Подумай

Виконай самостійне завдання

## Кольорові підказки

**Назва**  
**компонента**

**Назва**  
**властивості**

**Значення**  
**властивості**



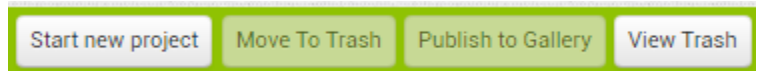
# Частина 1. Розробляємо дизайн

1

Відкрий середовище MIT App Inventor, яке знаходиться за посиланням <http://ai2.appinventor.mit.edu/>

2

Створи новий проєкт

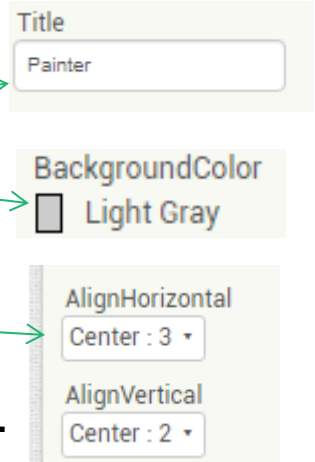


**Примітка. Давай своїм проєктам зрозумілі назви**

3

Зміни для екрану властивість:

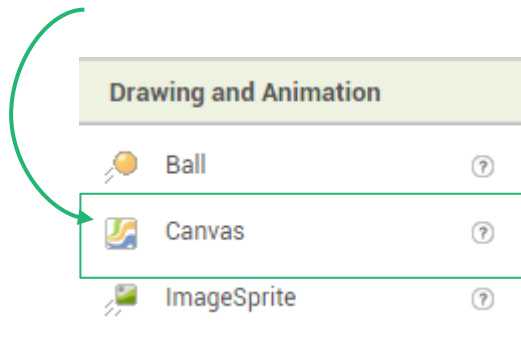
- **Title** на ім'я проєкту.
- **Background Color** (Колір фону) встанови **Light Gray**
- Властивості **Align Horizontal** та **Align Vertical** встанови по центру



Також, за бажанням, можеш змінити властивість **Theme**.

4

Додай на екран елемент **Canvas**. Він знаходиться в **Palette => Drawing and Animation**



**Canvas** - це

двовимірний сенсорний прямокутний панель, на якій можна малювати, а також розташовувати спрайти і текст.

Розташування об'єкта на Canvas задається як пара значень (X, Y), де X - кількість пікселів від лівого краю полотна, Y - кількість пікселів від верхнього краю полотна.



5

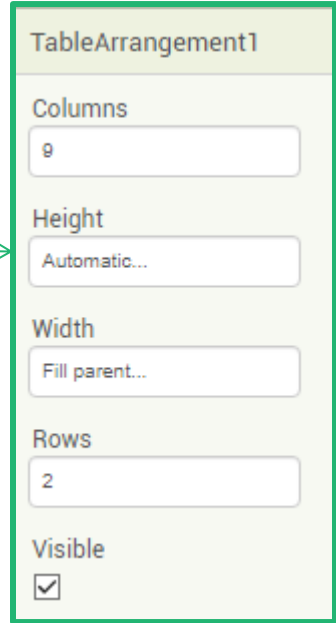
Зміни розмір для елемента **Canvas**.



6

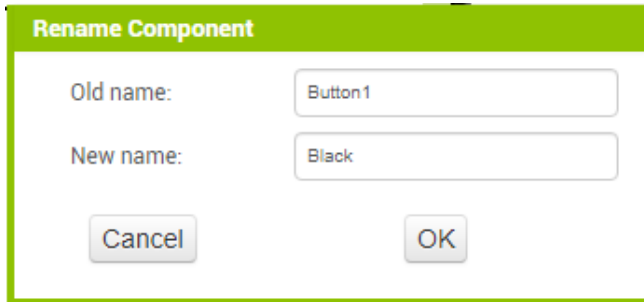
Щоб малюнок не був тільки одного кольору, додай можливість змінювати колір олівця. Для цього тобі потрібно додати декілька кнопок, але спершу додай на екран **Table Arrangement** та встанови для нього такі властивості.

Цей елемент тобі знадобиться для розташування кнопок із кольором.



7

Додай в середину **Table Arrangement** елемент **Button**. Зміни її ім'я на **black**.

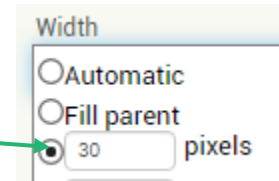


Зміни для неї такі властивості:

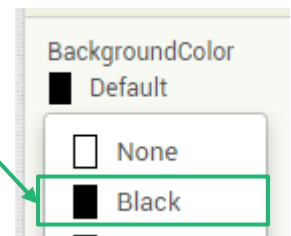
- Поле властивості **Text** залиш порожнім;



- ширину кнопки зроби рівною **30 пікселів**



- **Background Color** (Колір фону) вибери **Black**

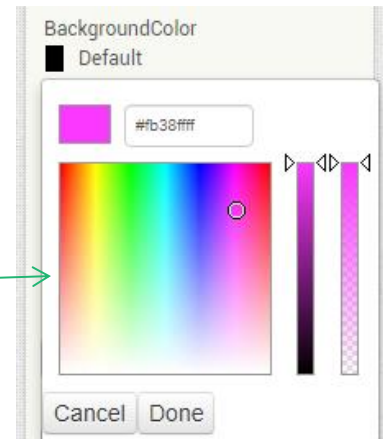


Аналогічно зроби палітру із 16 різних кольорів.

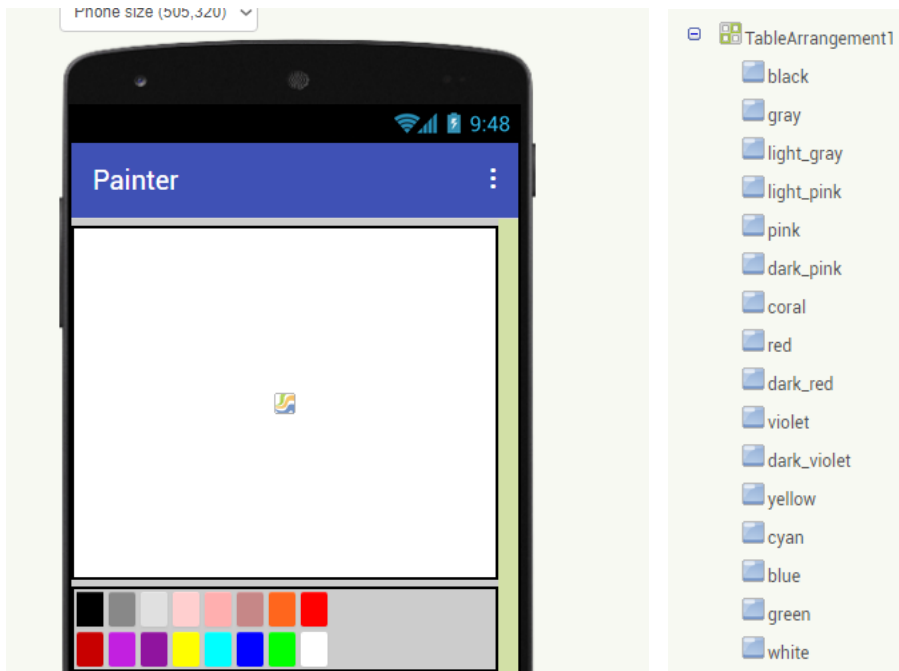
(по 8 в кожному рядку таблиці). Кольори можеш вибрати на власний розсуд.

**Примітка 1. Обов'язково давай кнопкам відповідну до їх кольорів назву.**

**Примітка 2. Щоб отримати більше кольорів, обери зі списку . Тоді тобі відкриється розширена палітра кольорів, яка виглядає так:**



**В тебе мало б вийти щось таке:**



**8** Додай можливість очищати поле для малювання. Для цього додай ще одну кнопку в середину **Table Arrangement**.

Зміни ім'я кнопки на **clear**, а властивість **Text** встанови **Очистити поле**

**9** Додай можливість змінити колір тла. Але цього разу зміна кольору не буде відбуватися за рахунок кнопок. Використай елемент

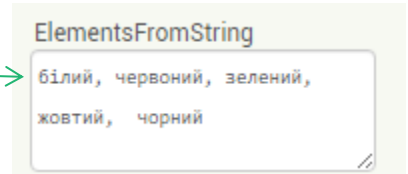


, який ти можеш знайти в групі **User Interface**.

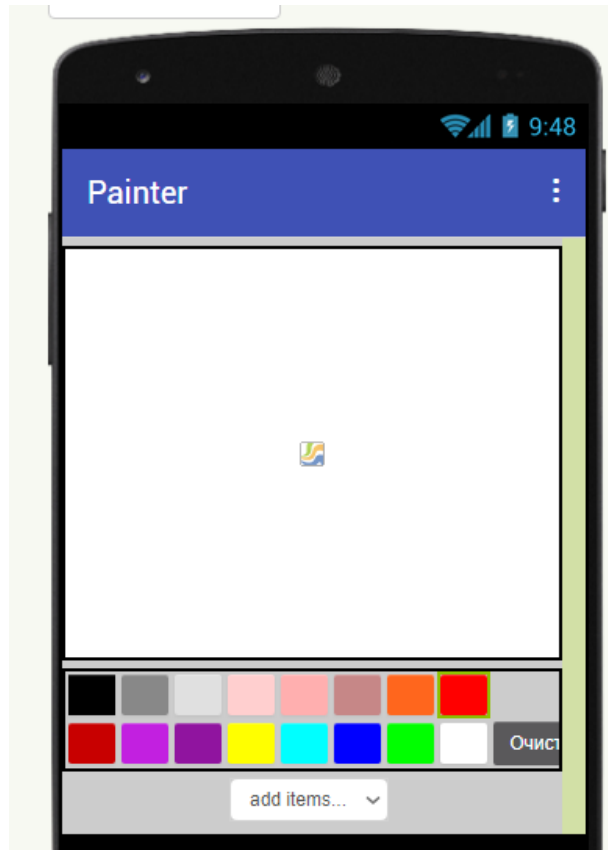
**Spinner** – це випадаючий список, тобто після того, як користувач натисне на цей елемент, йому відкриється список можливих варіантів.

Щоб додати до списку елементи, зміни властивість *Elements From String*.

**Примітка.** *Всі елементи списку розділюються комою.*



**Вигляд інтерфейсу:**





## Частина 2. Програмуємо елементи

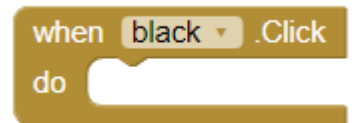
10

Перейди в режим розробника.



11

В меню **Blocks** для однієї з кнопок виберіть блок:

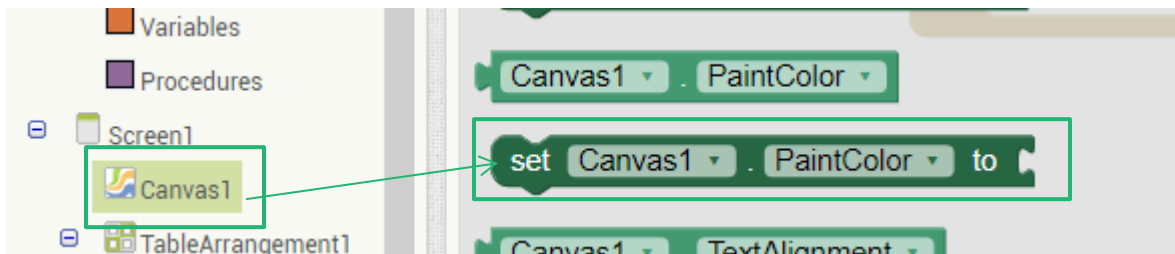


12

Коли ти будеш натискати на кнопку, колір олівця має змінюватись на колір даної кнопки.

За колір олівця відповідає поле **Paint Color** об'єкту **Canvas1**.

Додай в середину попереднього блоку такий блок:

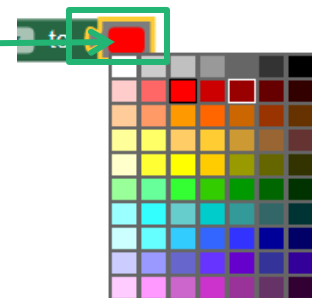


13

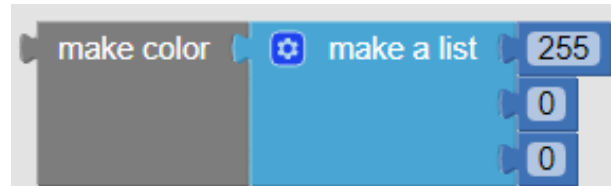
Подивись на утворений скрипт. Що ти бачиш? Помилку, чи не так?

Все через те, що програма не розуміє який саме колір олівця ти хочеш встановити.

У списку груп ти можеш побачити окрему групу **Colors**. Декілька блоків в цій групі – це вже готові кольори. Розкрити більшу палітру ти можеш натиснувши на колір в середині блоку.



Але ця палітра мала, тож тобі може знадобитися окрім готових кольорів також такий блок:



Кольори в цьому блоці – це кольори моделі RGB

Палітру кольорів RGB ти можеш знайти на цьому сайті: [https://www.w3schools.com/colors/colors\\_picker.asp](https://www.w3schools.com/colors/colors_picker.asp)

На цьому сайті ти можеш підбирати кольори як сам, так і використати перетворення з однієї моделі в іншу.

Для другого способу перейди в режим дизайнера та подивися на колір кнопки в

палетці (наприклад:  ).

RGB — адитивна колірна модель, що описує спосіб синтезу кольору, за якою червоне, зелене та синє світло накладаються разом, змішуючись у різноманітні кольори.



Or Enter a Color:

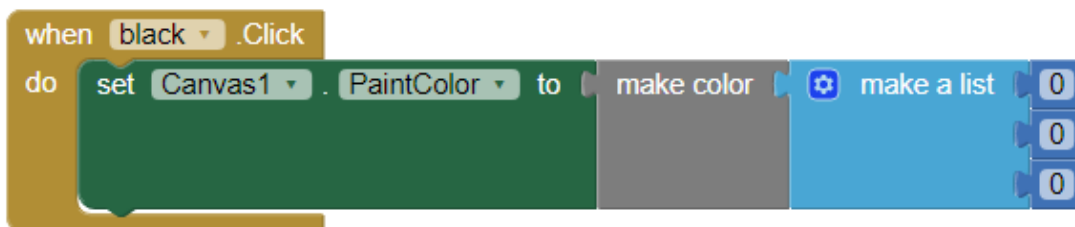
Скопіюй значення кольору та встав його в поле. Після чого, натисни кнопку **OK**.

Наприклад: програмуючи кнопку з чорним кольором олівця в тебе могли вийти два різні випадки:

1.




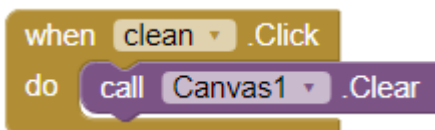
2.



**14** Повтори кроки №11-13 для того, щоб запрограмувати інші кнопки з кольорами.

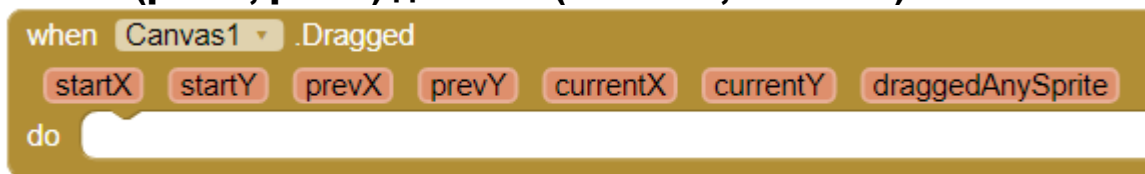
*Примітка. Для кожної кнопки уважно вибирай колір, який відповідатиме кольору заданому в дизайні.*

**15** Для очищення полотна використай блок  , який ти можеш знайти натиснувши **Blocks => Canvas1**.

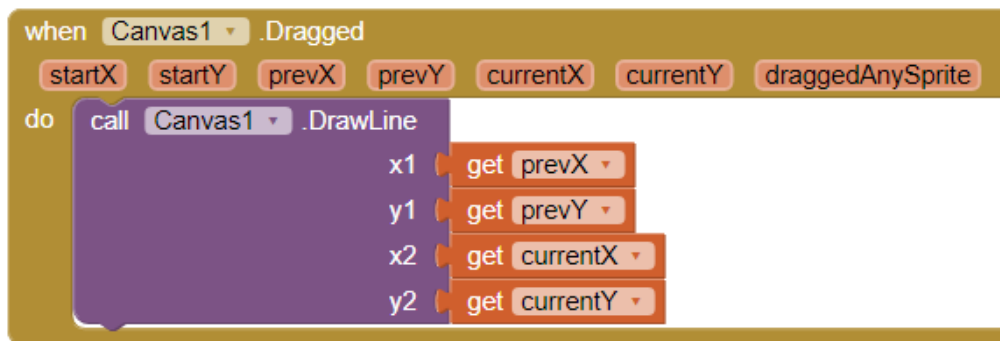


**16** опиши малювання лінії.

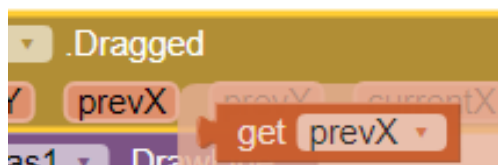
Для цього знайди блок **when Canvas1.Dragged**, він знаходиться в блоках **Canvas1**. Цей блок спрацьовує коли користувач виконує перетягування з однієї точки (prevX, prevY) до іншої (currentX, currentY).




**17** Для малювання лінія існує метод **callCanvas1.Drawline** . Лінія малюватиметься між точками (x1,y1) і (x2,y2).



*Примітка. Для того, щоб отримати змінні prevX, prevY, currentX та currentY, піднесіть курсор мишки до поля в блоці when Canvas1.Dragged*





**18** Опиши зміну кольору тла. Для цього обери блок **When tlo. After** **Selecting** з групи . Він спрацює після вибору користувачем одного з елементів списку.

```
when tlo .AfterSelecting
  selection
do
```

**19** Спробуй зібрати такий блок:

Перевірка  
вбраного  
користувачем  
елемента






```
if tlo . Selection = "білий"
then set Canvas1 . BackgroundColor to
```

Встановляє  
колір фону

**20** З'єднай попередній блок із **When tlo. After Selecting**

```
when tlo .AfterSelecting
  selection
do
  if tlo . Selection = "білий"
  then set Canvas1 . BackgroundColor to
```

**21** Але ж ти описав тільки один колір. Повтори два попередні кроки і для інших кольорів в списку

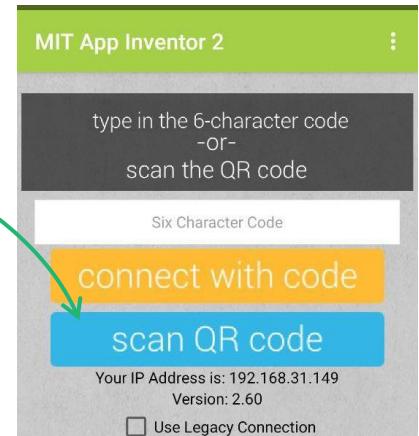
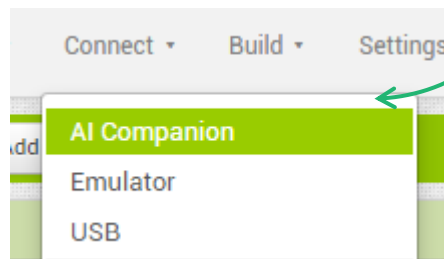
```
when tlo .AfterSelecting
  selection
do
  if tlo . Selection = "білий"
  then set Canvas1 . BackgroundColor to 
  if tlo . Selection = "червоний"
  then set Canvas1 . BackgroundColor to 
  if tlo . Selection = "зелений"
  then set Canvas1 . BackgroundColor to 
  if tlo . Selection = "жовтий"
  then set Canvas1 . BackgroundColor to 
  if tlo . Selection = "чорний"
  then set Canvas1 . BackgroundColor to 
```



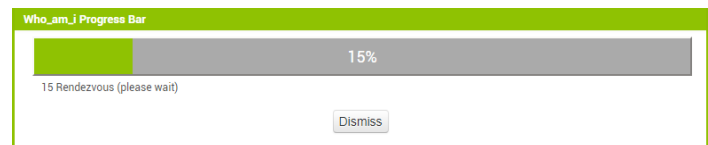
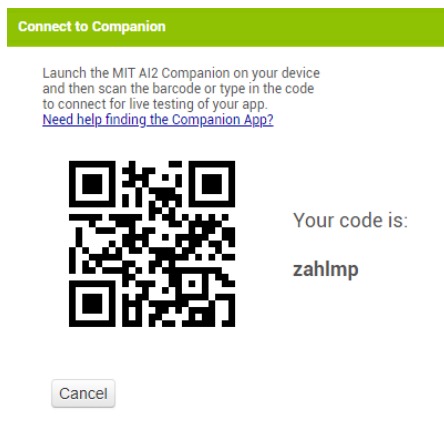
## Випробуй свій проект

**22** Відкрий на смартфоні додаток MIT AI2 Companion та натисни кнопку Scan QR code.

**23** Відкрий меню Connect в середовищі App Inventor та вибери зі списку



**24** Відскануй смартфоном QR код, який відкрився та зачекай завантаження твого додатку



**25** Випробуй свій додаток.

1. Чи всі кнопки вірно змінюють колір олівця?
2. Чи спрацьовує очищення тла?
3. Чи правильно змінюється колір тла?

Що буде в цьому проєкті?

Ти розробиш програму «Калькулятор». Особливістю цієї програми буде те, що користувач буде вводити числа за допомогою прописаних нами кнопок.

Навички, які ти отримаєш

Поглибиш свої знання змінних. Дізнаєшся, що таке процедури, як їх створювати та використовувати.

## Умовні позначення



Виконай

Виконуй ці ІНСТРУКЦІЇ одну за одною



Випробуй

Надішли програму на свій телефон за допомогою QR коду



Подумай

Виконай самостійне завдання

## Кольорові підказки

**Назва**  
**компонента**

**Назва**  
**властивості**

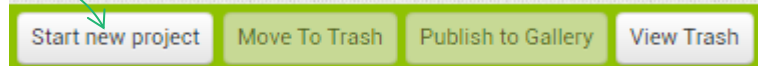
**Значення**  
**властивості**



# Частина 1. Розробляємо дизайн

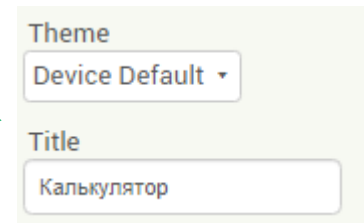
**1** Відкрий середовище MIT App Inventor, яке знаходиться за посиланням <http://ai2.appinventor.mit.edu/>

**2** Створи новий проєкт



**Примітка. Давай своїм проєктам зрозумілі назви**

**3** Зміни властивості **Title** та **Theme** для екрану



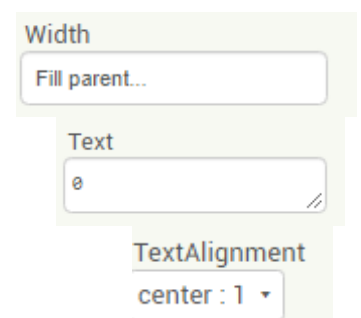
**4** Додай на екран компонент **Vertical Arrangement**.  
Зміни для нього такі властивості:



**5** Перетягни в **Vertical Arrangement** компонент **Label**. Перейменуй його на **result**

**6** Зміни властивості компонента **result**.

- Властивість **Width** зроби рівною **батьківському компоненту**;
- Властивість **Text** зроби рівною **0**;
- Властивість **Text Alignment** зроби **по центрі**;



- Властивість **Font Size** зроби рівною **70**;
- Для властивість **FontTypeface** встанови значення **monospace**.

FontSize  
70

FontTypeface  
monospace ▾

7

Перетягни в **Vertical Arrangement** компонент **Table Arrangement** та зміни такі його властивості:

Rows  
3

Columns  
4

Height  
Automatic...

Width  
Fill parent...

8

Тепер додай кнопки, які будуть відповідати за цифри та операції.

1	2	3	+
4	5	6	=
7	8	9	C

Помісти в контейнер **Table Arrangement** кнопку та перейменуй її на **n1**.

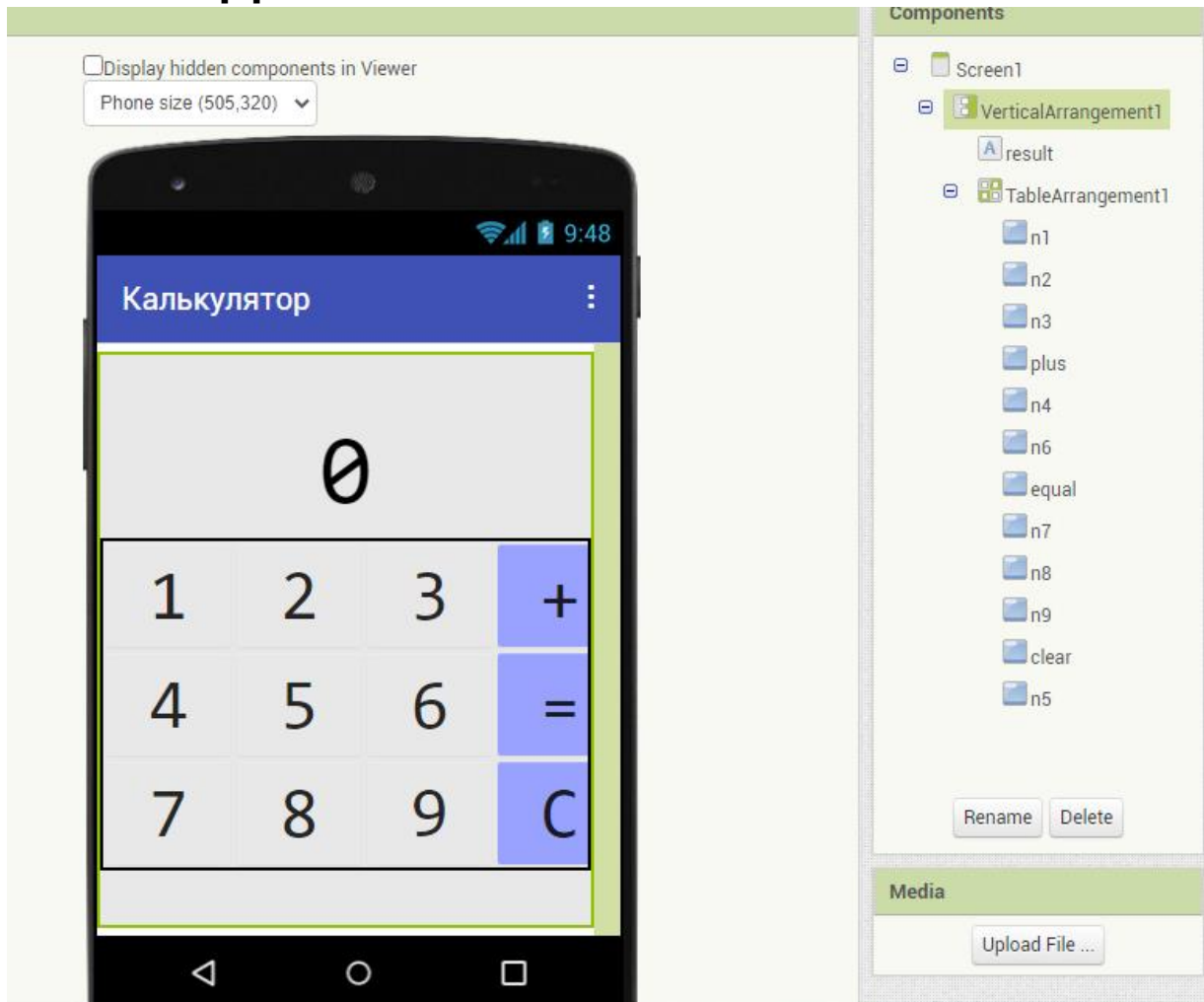
Зміни для кнопки такі властивості:

- + **Background Color** встанови **None** → BackgroundColor  
None
- + **Font Size** зроби рівним **50** → FontSize  
50
- + Для **Font Typeface** обери **monospace** → FontTypeface  
monospace ▾
- + **Width** встанови **25%** → Width  
25 percent...
- + **Text** зміни на **1** → Text  
1

**Аналогічно додай інші кнопки.**

**Примітка.** Для властивості **Background Color** кнопок із операціями встанови який-небудь колір. Цим самим ти візуально відділиш їх від кнопок із цифрами.

**Готовий інтерфейс:**



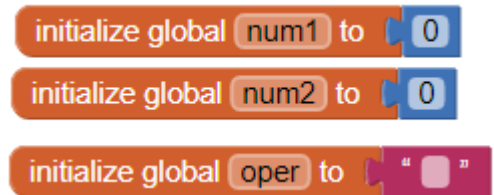


## Частина 2. Програмуємо елементи

- 9 **Перейди в режим розробника.**

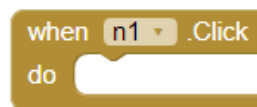


- 10 **Ініціалізуй три змінні – дві числового типу та одну текстового. Змінні `num1` та `num2` будуть зберігати в собі числа, над якими виконується операція.**



**Змінна `oper` необхідна для того, щоб дізнатись, яка саме математична операція буде виконуватись над числами.**

- 11 **Додай для кнопки `n1` блок**



- 12 **Коли користувач вводить яку-небудь цифру, він розуміє, що остання стає на місце одиниць, а розряд попереднього числа - збільшується.**

**Наприклад:** На екрані є вже число **123** (1 – сотні, 2 – десятки, 3 – одиниці). Якщо користувач введе ще 4, то це число зміниться на **1234**. (1 – тисячні, 2 – сотні, 3 – десятки, 4 – одиниці)

**Але комп'ютер – не людина. Він не вмє думати, зате він може виконувати певні потрібні нам дії.**

**Для того, щоб комп'ютер розумів, з якими саме операндами йому потрібно працювати, склади такий скрип:**



**Переприсвоїть змінній `num1` нове значення**

**Збільшить розряд попереднього числа**

**Збільшить отримане число на необхідну кількість одиниць**



13

Зміни напис **result** на отримане число.

```
set result . Text to get global num1
```

14

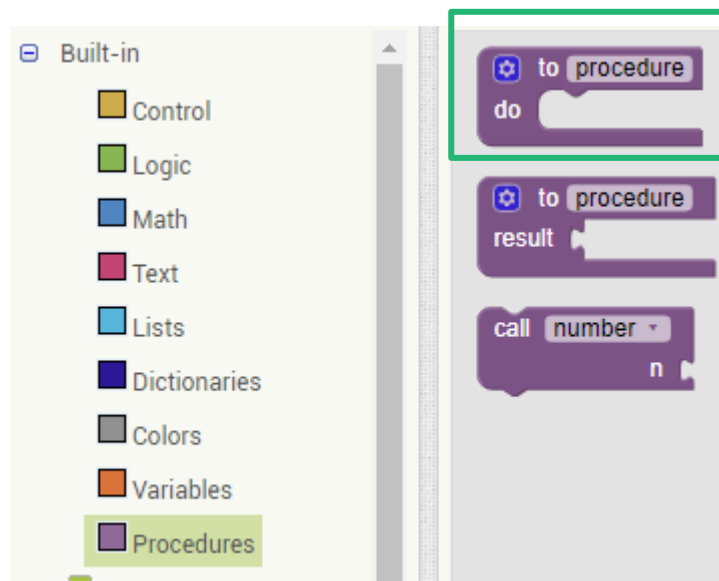
З'єднай блоки з попередніх трьох кроків утворивши такий скрипт:

```
when n1 .Click  
do  
  set global num1 to get global num1 * 10 + 1  
  set result . Text to get global num1
```

15

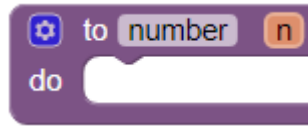
Щоб калькулятор запрацював, потрібно створити такі ж програми і для інших кнопок. У твоєму проєкті їх усього 9, проте, в іншому їх може бути набагато більше. Що ж можна зробити, щоб не створювати щоразу довгу програму? Можна скористатися процедурами!

Процедура — це особливий блок, за допомогою якого можна скоротити обсяг програми



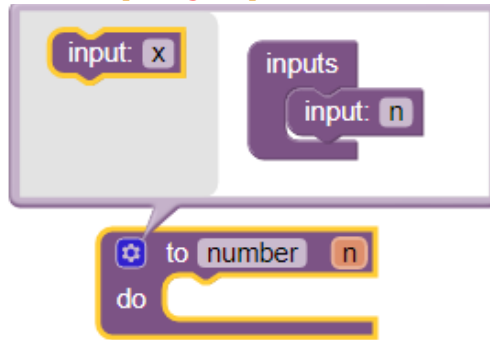
16

Створи процедуру **number**. Додай до процедури нову вхідну (**input**) з назвою **n**.



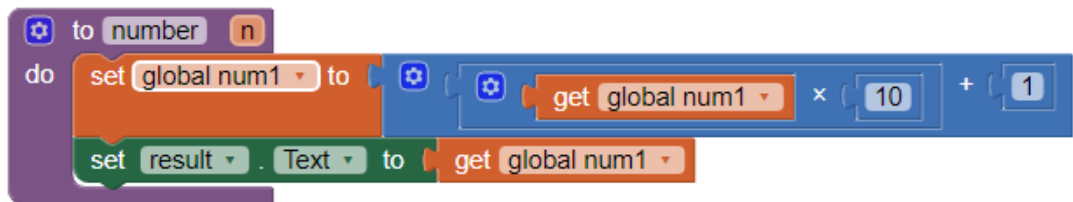
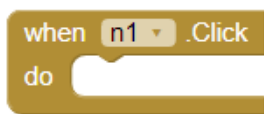
*Примітка 1. Вхідна **n** буде рівна цифрі, для якої викликається процедура.*

*Примітка 2. Для того, щоб отримати доступ до додаткових опцій, натисни на шестірню і перенеси блок **input** у **inputs***



17

Сама процедура відповідає за перетворення числа після того, як користувач вводить останню цифру. Отже, перенеси вміст із блоку **when n1.Click** в блок **to number**.



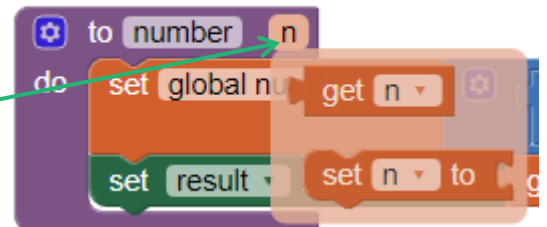
18

Для кожної кнопки ця процедура має працювати по різному. Чи не так?

Різнитися буде тільки цифра, яка додається до числа, отже заміни блок



*Примітка. Блок **get n** заховано в самій процедурі. Щоб його отримати, наведи курсор миші на назву вхідної*



Готовий скрипт виглядає так:

```
to number n
do
  set global num1 to (get global num1 * 10 + get n)
  set result . Text to get global num1
```

19

Опиши виклик процедури для кожної з кнопок

```
when n1 .Click
do
  call number n 1

when n2 .Click
do
  call number n 2

when n3 .Click
do
  call number n 3

when n4 .Click
do
  call number n 4

when n5 .Click
do
  call number n 5

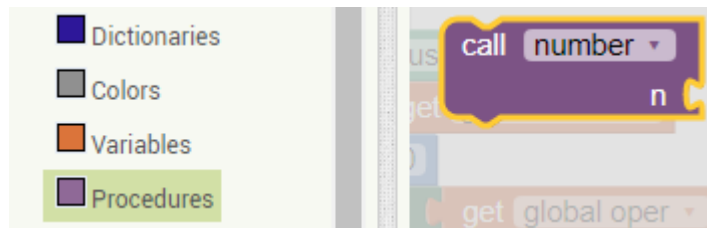
when n6 .Click
do
  call number n 6

when n7 .Click
do
  call number n 7

when n8 .Click
do
  call number n 8

when n9 .Click
do
  call number n 9
```

**Примітка.** Блок call number ти можеш знайти у групі Procedures



20

Запрограмуй кнопку plus. Додай блок

Додай всередину цього блоку:

1. Скрипт `set global oper to plus . Text`, щоб комп'ютер зрозумів, яку саме операцію користувач обрали.
2. Скрипт `set result . Text to get global oper`, щоб користувач впевнився в тому, що він обрав правильну операцію
3. Скрипт `set global num2 to get global num1`. Він прирівняє другий операнд до першого. Це потрібно для того, щоб скоротити програму

та не створювати іншу процедуру для обробки другого числа.

4. Скрипт  
рівним 0.

```
set global num1 to 0
```

```
when plus .Click  
do  
  set global oper to plus .Text  
  set result .Text to get global oper  
  set global num2 to get global num1  
  set global num1 to 0
```

Готовий скрипт:

21

Запрограмуй кнопку equal. Додай блок

```
when equal .Click  
do
```

Додай перевірку операції та обчислення суми.

```
if  
  get global oper = "+"  
then  
  set global num2 to get global num1 + get global num2
```

Додай блоки.

```
set result .Text to get global num2  
set global num1 to get global num2  
set global num2 to 0
```

Готова програма:

```
when equal .Click  
do  
  if  
    get global oper = "+"  
  then  
    set global num2 to get global num1 + get global num2  
  set result .Text to get global num2  
  set global num1 to get global num2  
  set global num2 to 0
```

22

Опиши кнопку clear

```
when clear .Click  
do  
  set global num1 to 0  
  set global num2 to 0  
  set result .Text to "0"
```



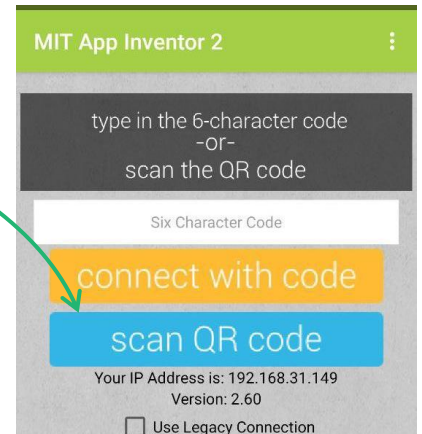
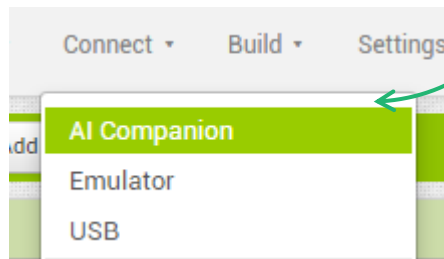
## Випробуй свій проект

23

Відкрий на смартфоні додаток **MIT AI2 Companion** та натисни кнопку **Scan QR code**.

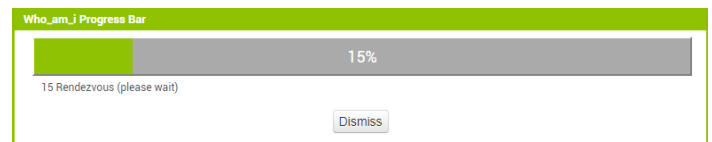
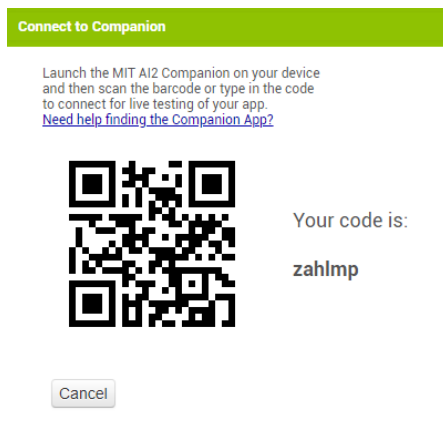
24

Відкрий меню **Connect** в середовищі **App Inventor** та вибери зі списку



25

Відскануй смартфоном QR код, який відкрився та зачекай завантаження твого додатку



26

Випробуй свій додаток.

1. Чи правильно рахує твій калькулятор?
2. Чи є можливість додати нове число одразу отриманої попередньої суми?



## Самостійне завдання

### Спробуй виконати\*

Додай до калькулятора операції віднімання, множення та ділення.  
Щоб це реалізувати додай на екран ще три кнопки.

**Примітка 1.** Щоб помістити додаткові кнопки в **Table Arrangement** зміни властивість **Columns** або **Rows**

Що буде в цьому проєкті?

Гра «Клікер».  
Суть цієї гри: натиснути на  
об'єкт якомога більше разів за  
визначений час.

Навички, які ти отримаєш

Навчишся створювати  
багатоекранні додатки.  
Дізнаєшся про компонент  
Image Sprite.

## Умовні позначення



**Виконай**

**Виконуй ці ІНСТРУКЦІЇ одну за одною**



**Випробуй**

**Надішли програму на свій телефон за допомогою QR коду**



**Подумай**

**Виконай самостійне завдання**

## Кольорові підказки

**Назва  
компонента**

**Назва  
властивості**

**Значення  
властивості**

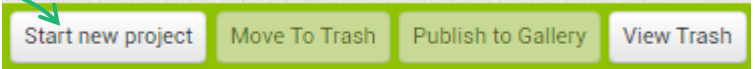


## Частина 1. Розробляємо дизайн

**1** Завантаж файл *clicker.jpg* на свій комп'ютер. Це зображення – це об'єкт, який буде переміщуватися по екрані.

*Примітка. Зберігай файли у спеціально створеній для проєкту папці.*

**2** Відкрий середовище MIT App Inventor, яке знаходиться за посиланням <http://ai2.appinventor.mit.edu/>

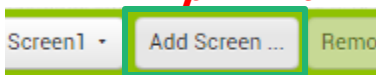
**3** Створи новий проєкт 

*Примітка. Давай своїм проєктам зрозумілі назви*

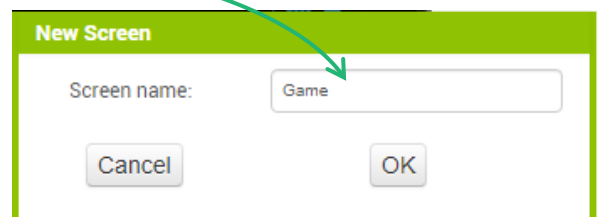
**4** В цьому проєкті буде три різних екрани:  
1. Початковий екран із меню  
2. Екран із грою  
3. Екран із результатами.

Екран для меню ти вже маєш . Два інші тобі потрібно створити.

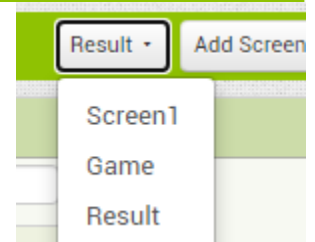
*Примітка Для того, щоб створити екран, натисни*



*, Після чого, введи назву екрану*



Після створення, назва екрану з'явиться в переліку екранів.

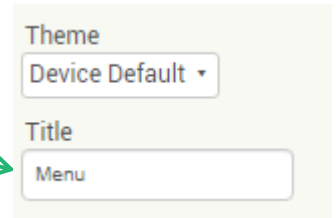




5

Налаштуй початковий екран.

Зміни для нього властивості **Title** та **Theme** для екрану



6

Додай на екран **напис**, щоб користувач бачив назву гри.

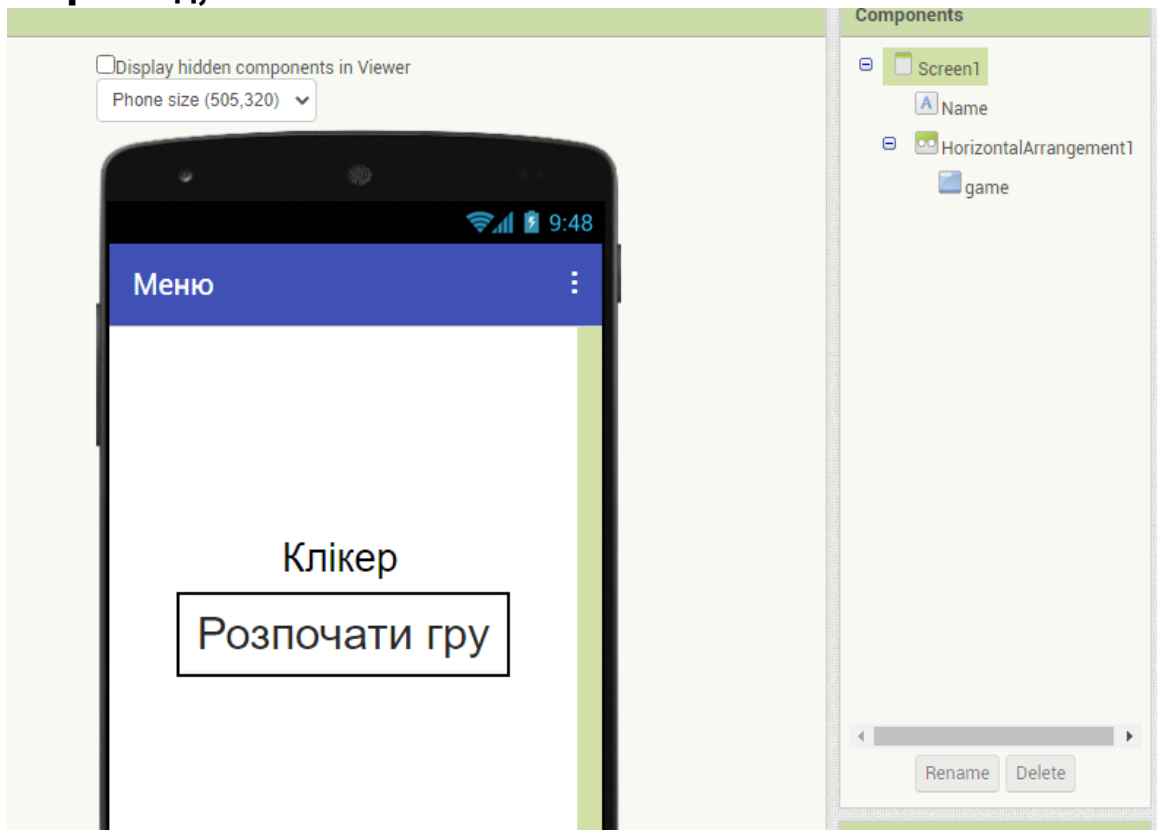
7

Додай на екран **кнопку**, яка запустить гру.

8

Розташуй елементи на екрані в бажаному порядку.

Наприклад, в тебе могло б вийти таке:



9

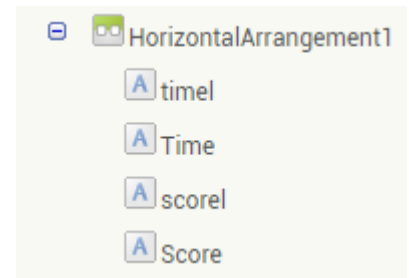
Перейди на екран **Game**



**10** Додай на екран компонент **Horizontal Arrangement**. В ньому будуть міститися поля із кількістю очок та часом.

**11** В середину контейнера додай 4 елементи напису.

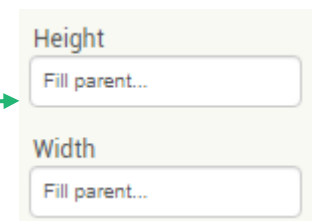
Залишилось часу: --- Очки: ---



**12** Додай на екран компонент **Canvas**. В межах цього поля ти можеш зробити переміщення об'єкта.

Canvas

Встанови для нього розмір:



**13** Поле в тебе вже є, а самого об'єкта - немає. Додай на поле елемент **Image Sprite**.

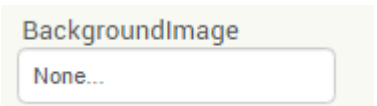
ImageSprite

**Примітка. Обов'язково помісти цей елемент в Canvas.**

Встанови для цього елемента розмір

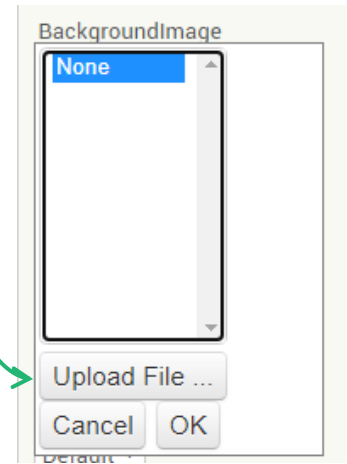
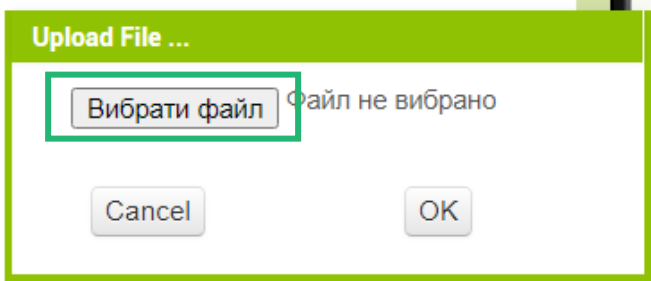


**14** Встанови зображення для елемента **Image Sprite**. Для цього натисни на значення **None** властивості **Background Image** із списку доступних для компонента **Image Sprite**.



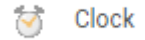
Натисни на кнопку **Upload File**

У діалоговому вікні натисни **Вибрати файл** та знайди на комп'ютері файл **clicker.jpg**.

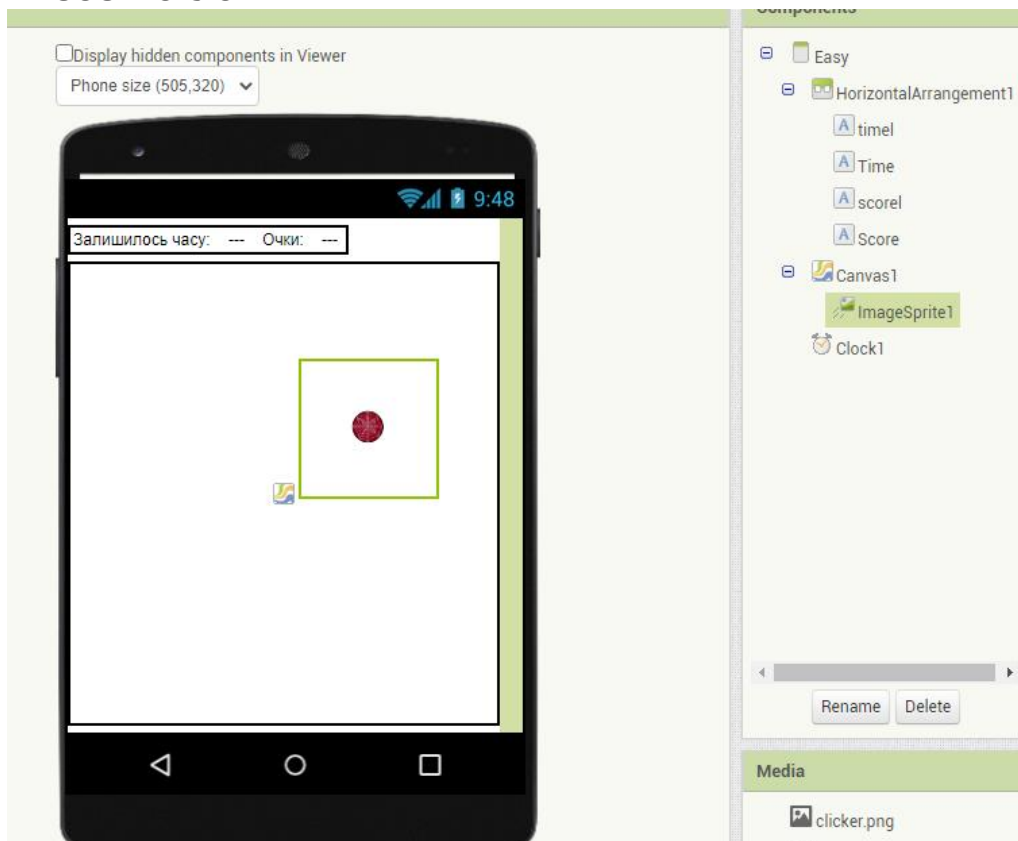


15

Останнім елементом на цьому екрані буде **Clock**, який ти можеш знайти в групі **Sensors**. За допомогою нього буде здійснюватися відлік часу.



Ось таке в тебе могло вийти:

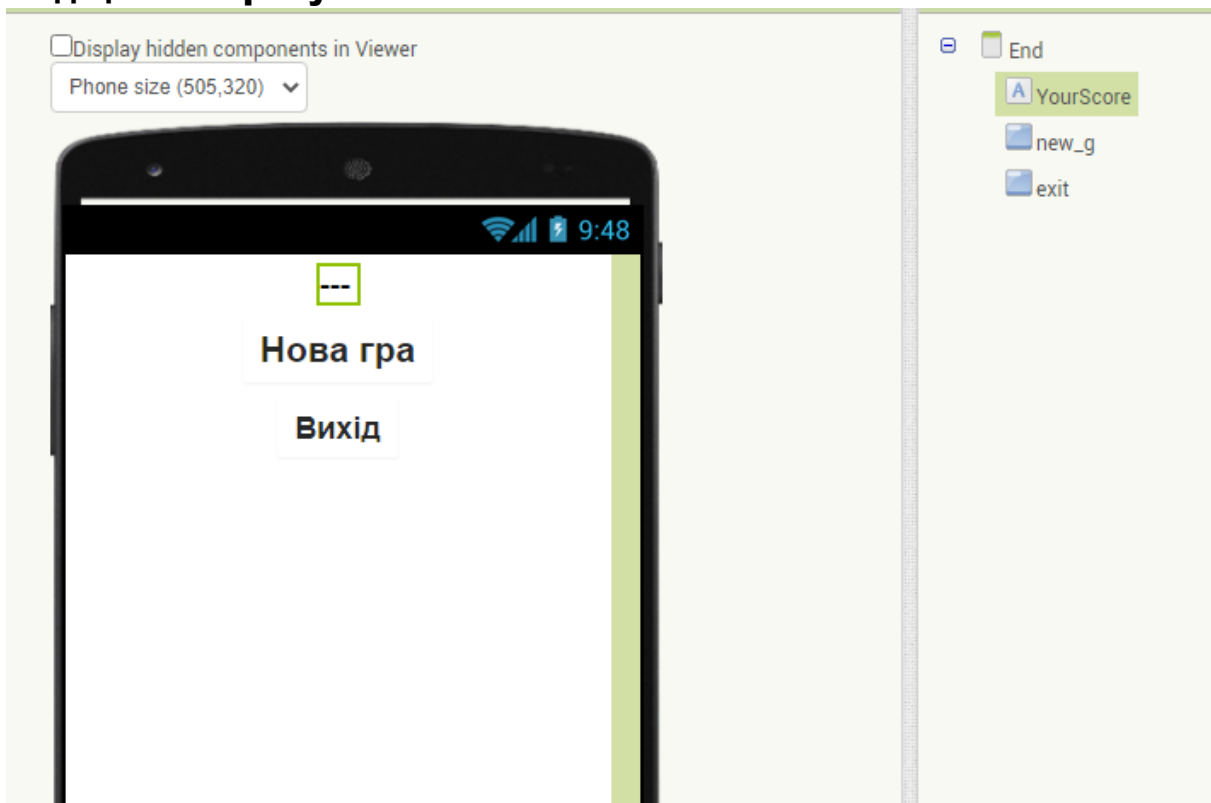


**16** Перейди на результативний екран



**17** Додай на цей екран напис **Your Score**, в якому буде відображатись результат проходження гри.

**18** Додай на екран дві кнопки – «Нова гра» та «Вихід».  
Вигляд цього екрану:





## Частина 2. Програмуємо елементи

19

Перейди в режим розробника.



20

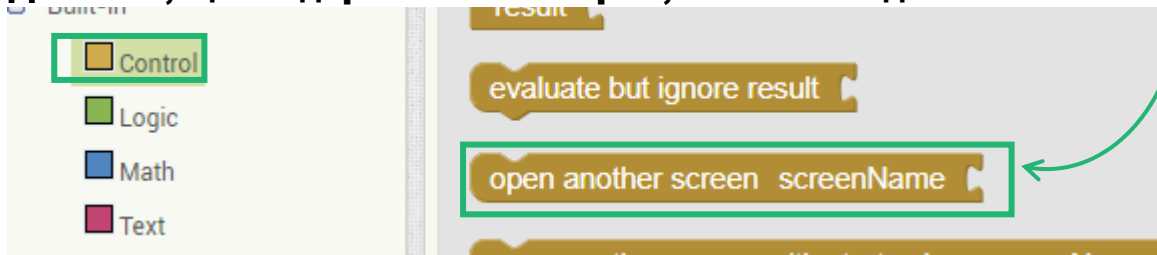
Так само як і в режимі дизайнера, ти можеш вибрати зі списку усіх екранів тільки один та скласти код для нього. Почни з Меню.

На цьому екрані в тебе є тільки одна кнопка. Натиснувши на неї, повинен відкритися інший екран з грою.



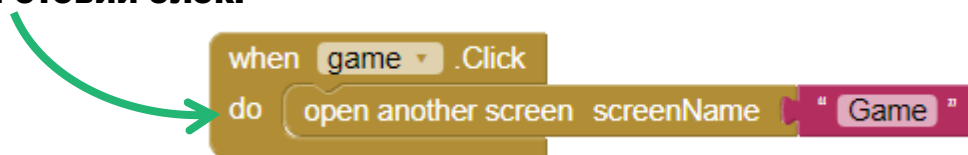
Додай блок на поле.

Для того, щоб відкрити інший екран, тобі необхідний такий блок



До цього блоку ти маєш прикріпити назву потрібного екрану.

Готовий блок:



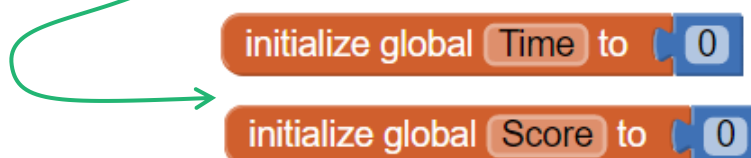
21

Перейди на екран **Game**



22

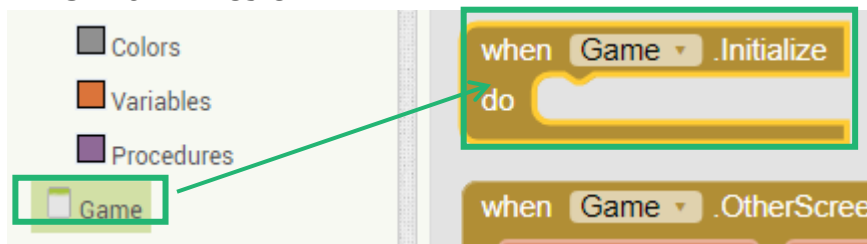
На цьому екрані тобі потрібні дві змінні: час та кількість балів. Додай для цього такі блоки:



23

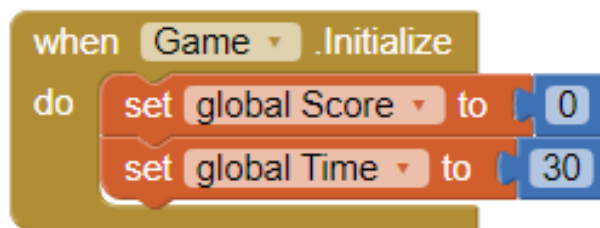
Кожного разу після того, як відкриється цей екран, кількість балів повинна дорівнювати нулю, а залишений час – 30 секунд.

Тобі знадобиться такий блок:



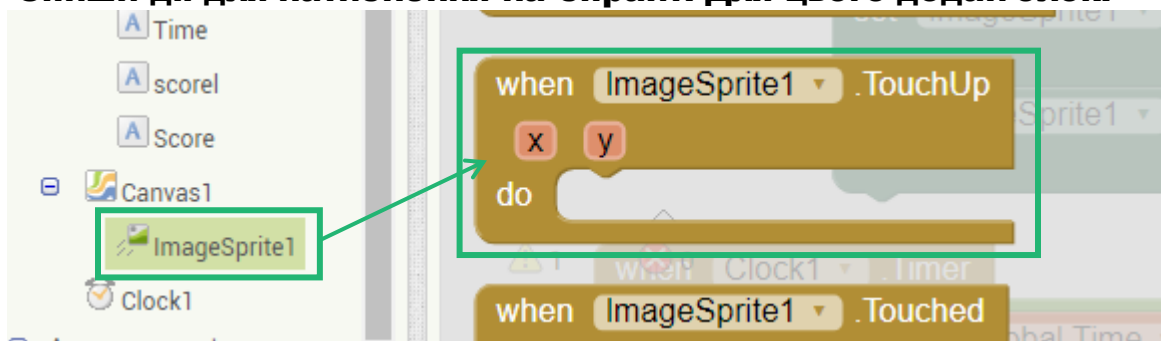
Він спрацює одразу ж після відкриття цього екрану.

Додай інші необхідні блоки



24

Опиши дії для натиснення на спрайт. Для цього додай блок:



Додай в середину:



- Скрипт збільшити кількість балів, щоб



- Скрипт після нарощення, оновляти текстове поле на екрані, щоб кожного разу

25

Додай можливість спрайта переміщуватись у випадкове місце на екрані. Для цього збери такий скрипт:

```

set ImageSprite1 . X to random integer from 1 to Canvas1 . Width - ImageSprite1 . Width
set ImageSprite1 . Y to random integer from 1 to Canvas1 . Height - ImageSprite1 . Height

```

Перший рядок змінює координати спрайта по горизонталі, а другий – по вертикалі.

Готовий блок виглядає так:

```

when ImageSprite1 . TouchUp
do
  set global Score to get global Score + 1
  set Score . Text to get global Score
  set ImageSprite1 . X to random integer from 1 to Canvas1 . Width - ImageSprite1 . Width
  set ImageSprite1 . Y to random integer from 1 to Canvas1 . Height - ImageSprite1 . Height

```

**26** Залишилось описати умову кінця гри.

У гравця на гру є лише 30 секунд.

Знайди в списку доступних для компонента **Clock** такий блок:

```

when Clock1 . Timer
do

```

Цей блок запускати вкладену програму кожної секунди.

**27** Тут тобі потрібно зробити перевірку. Якщо в гравця не залишилось часу, тоді відкривається інший екран

```

if get global Time = 1
then
  set Clock1 . TimerEnabled to false
  open another screen with start value screenName "End"
  startValue get global Score

```

**Примітка.** Цього разу блок для відкриття екрану трішки інший. Змінні одного екрану є невидимими для іншого. Цим блоком ти можеш передавати значення між екранами.

**28** Додай скрипт для зменшення лічильника часу

```

set global Time to get global Time - 1
set Time . Text to get global Time

```

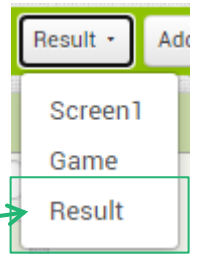
Готовий блок:

```

when Clock1.Timer
do
  if get global Time = 1
  then
    set Clock1.TimerEnabled to false
    open another screen with start value screenName "End"
    startValue get global Score
  set global Time to get global Time - 1
  set Time.Text to get global Time

```

**29** **Перейди на результативний екран**



**30** **Склади для цього екрану такі скрипти:**

```

when End.Initialize
do
  set YourScore.Text to get start value

```

**Встановить текст  
напису на  
кількість балів**

```

when new_g.Click
do
  open another screen screenName "Game"

```

**Відкриє екран з  
грою**

```

when exit.Click
do
  close application

```

**Вихід з програми**

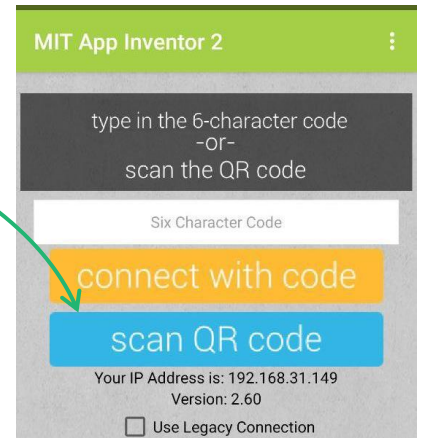
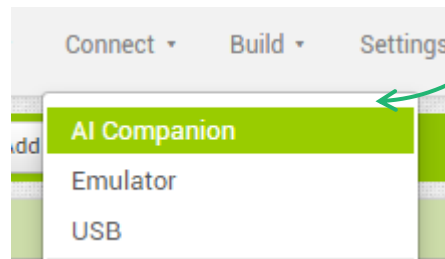




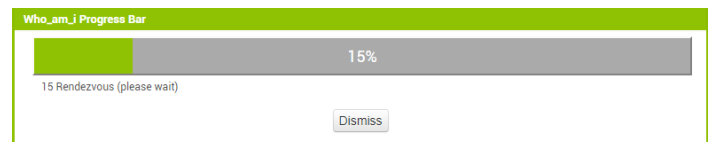
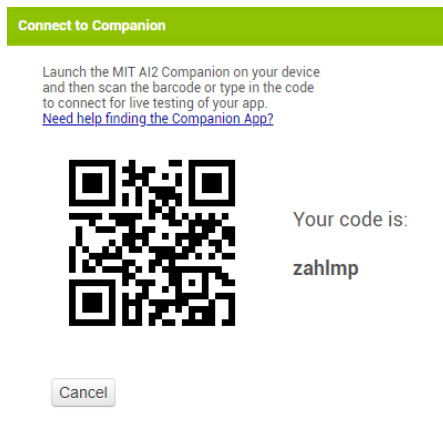
## Випробуй свій проект

**31** Відкрий на смартфоні додаток **MIT AI2 Companion** та натисни кнопку **Scan QR code**.

**32** Відкрий меню **Connect** в середовищі **App Inventor** та вибери зі списку



**33** Відскануй смартфоном QR код, який відкрився та зачекай завантаження твого додатку



**34** Випробуй свій додаток. Поділись з іншими своїм результатом проходження гри.



## Самостійне завдання

Оформи свою гру змінивши властивості елементів.

### Спробуй виконати\*

Додай можливість обрання рівня гри. Для цього додай до меню кнопки із назвами складності. Наприклад «Простий режим» та «Складний режим».

*Примітка. Рівні відрізняються розміром елемента Image Sprite.*

Що буде в цьому проєкті?

Створення програми «Довідник» з використанням списку.

Навички, які ти отримаєш

Ти дізнаєшся про розширені можливості списків та навчишся використовувати їх в своїй програмі.

## Умовні позначення



Виконай

Виконуй ці ІНСТРУКЦІЇ одну за одною



Випробуй

Надішли програму на свій телефон за допомогою QR коду



Подумай

Виконай самостійне завдання

## Кольорові підказки

**Назва**  
**компонента**

**Назва**  
**властивості**

**Значення**  
**властивості**



## Частина 1. Розробляємо дизайн

1

Створи на комп'ютері папку з назвою «Довідник».

Вибери 10 міст України, які захоплюють тебе найбільше, та завантаж їх герби в цю папку.

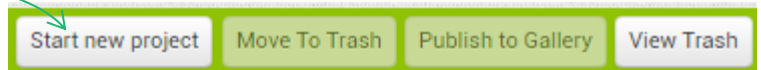
*Примітка. Тематику довідника можеш змінити. Наприклад: «Моделі машин», «Книги», «Рослини» тощо.*

2

Відкрий середовище MIT App Inventor, яке знаходиться за посиланням <http://ai2.appinventor.mit.edu/>

3

Створи новий проєкт



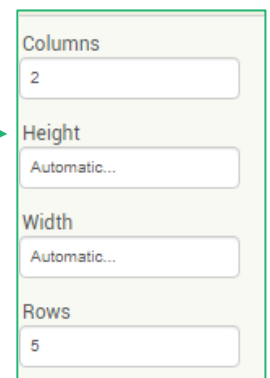
*Примітка. Давай своїм проєктам зрозумілі назви*

4

На початковому екрані в тебе буде розміщуватись меню. Тут користувач обиратиме місто, про яке він хоче дізнатись.

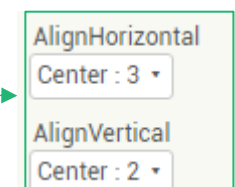
Додай на екран контейнер **Table Arrangement**, та встанови для нього такі властивості:

- Кількість колонок - 2
- Кількість рядків – 5
- Ширина та висота – автоматичний.



5

Встанови для екрану властивості **Align Horizontal** та **Align Vertical** по центрі.



6

Додай в середину контейнера **кнопки**.

Кожна кнопка – це окреме місто.

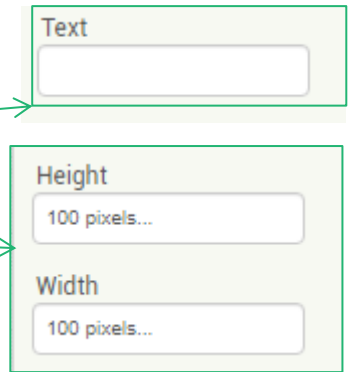
Встанови для кнопки властивість **Image**, завантаживши одне із заготовлених зображень. Ця властивість встановляє фонове зображення

**кнопки.**

**Перейменуй кнопку відповідно до міста.**

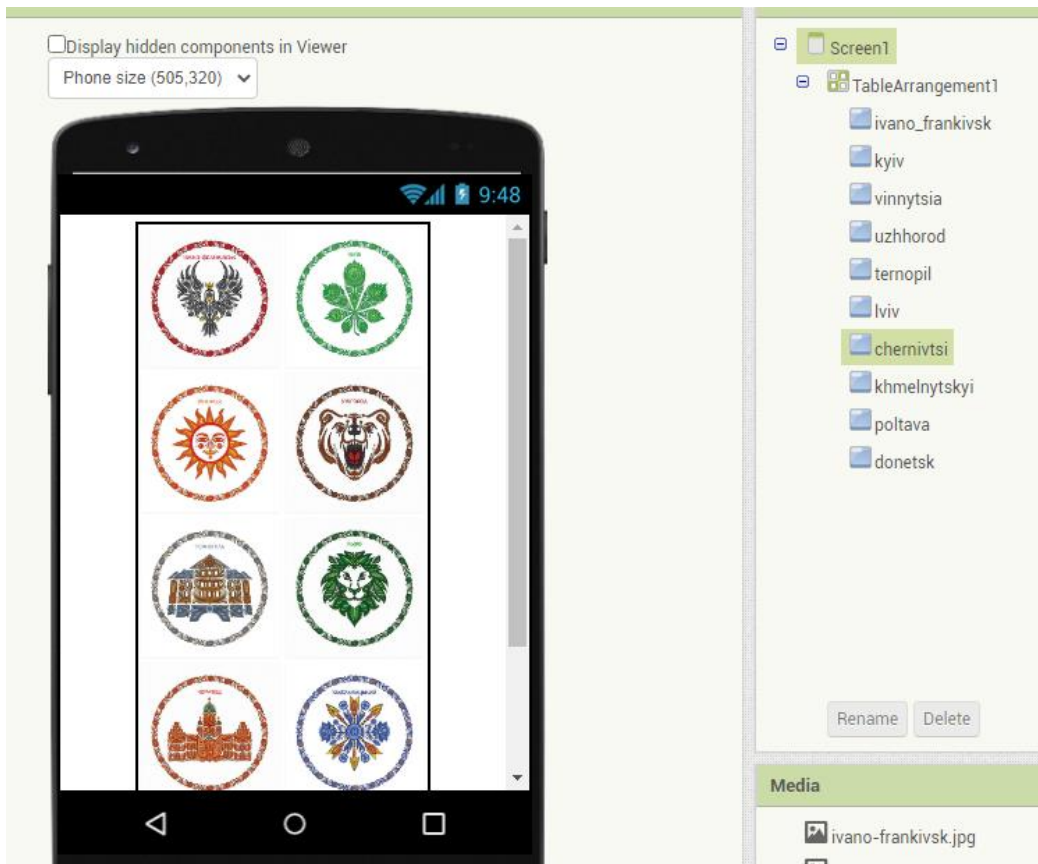
**Встанови інші властивості:**

- **Очисти текст кнопки, залишивши поле властивості **Text** порожнім;**
- **Встанови розміри кнопки – **Height** та **Width** рівні 100 пікселів.**



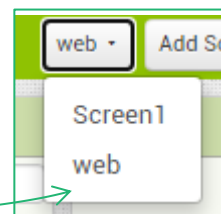
**Аналогічно встанови ці властивості для інших кнопок.**

**Кінцевий вигляд цього екрану:**



**7**

**Створи новий екран. Дай йому назву **Web**.  
Перейди на цей екран.**



**8**

**На цьому екрані буде відкриватись інтернет сторінка з інформацією**

**про обране місто. Для цього, тобі необхідний елемент**

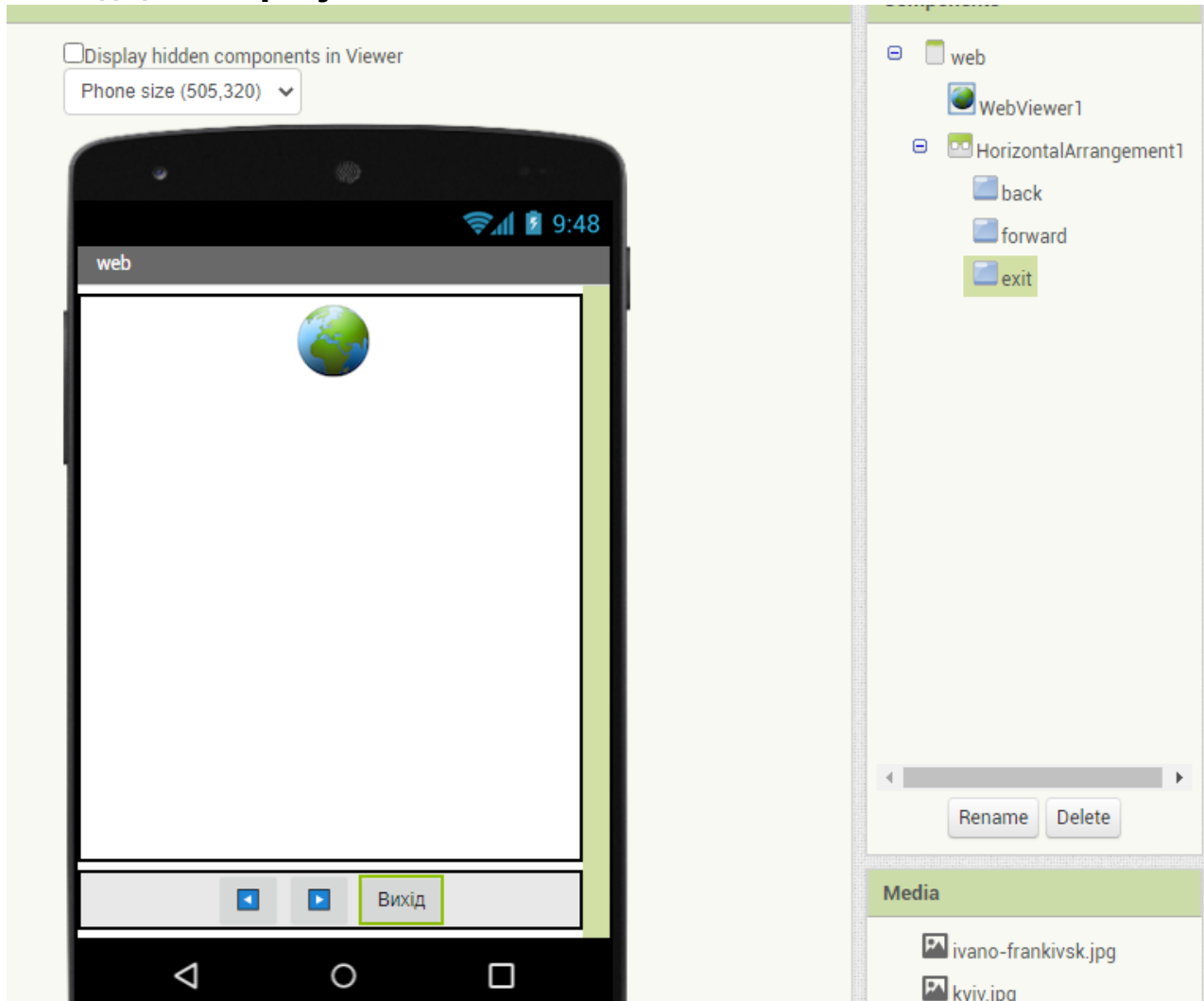


9

**Додай мінімальний функціонал браузера, а саме:**

- **Кнопку «Вперед»;**
- **Кнопку «Назад»;**
- **Кнопку «Вихід».**

**Вигляд цього екрану:**



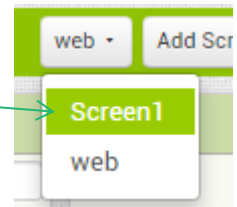


## Частина 2. Програмуємо елементи

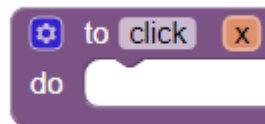
10 **Перейди в режим розробника.**



11 **Розпочни програмування з початкового екрану.**

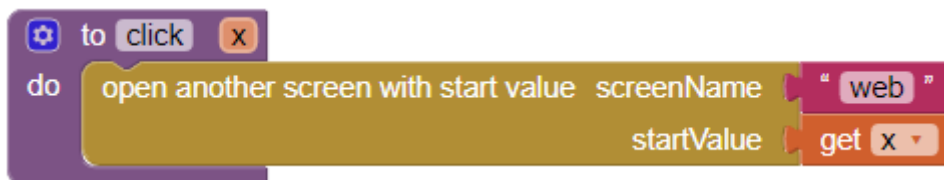


12 **Створи нову процедуру `click` з одним параметром.**

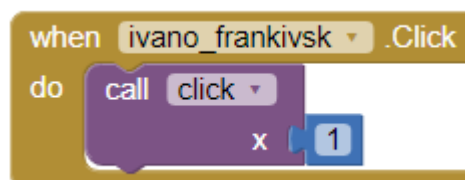


13 **Процедура `click` повинна відкривати екран, на якому міститься браузер.**

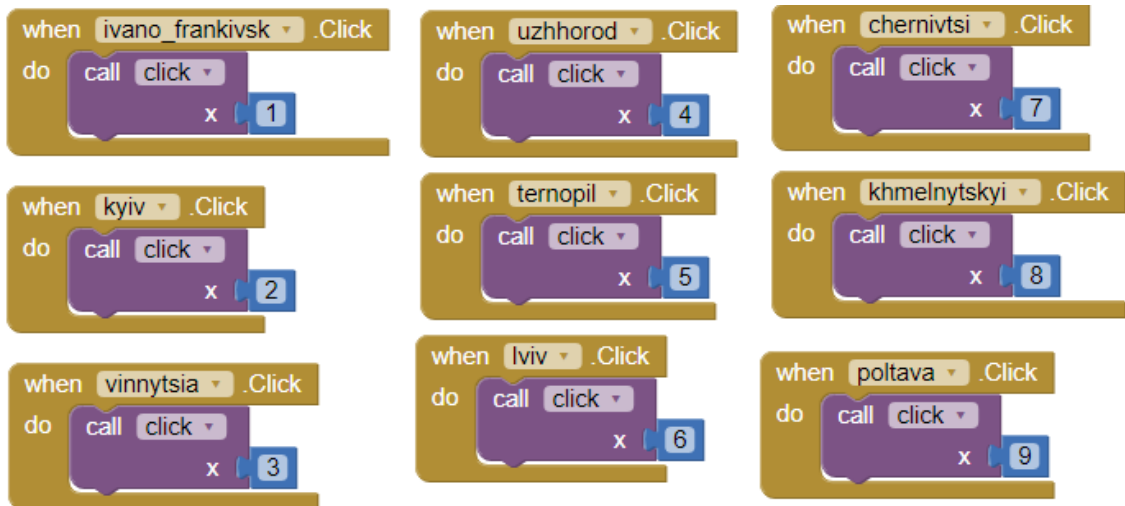
**Параметр `x` буде визначати інформацію, яку потрібно відкрити.**



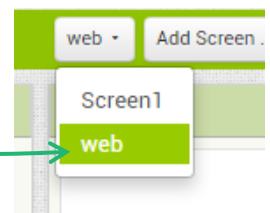
14 **Опиши виклик процедури для кожної кнопки.**



**Примітка. Для кожної кнопки параметр `x` – різний**



**15** **Перейди на екран web**

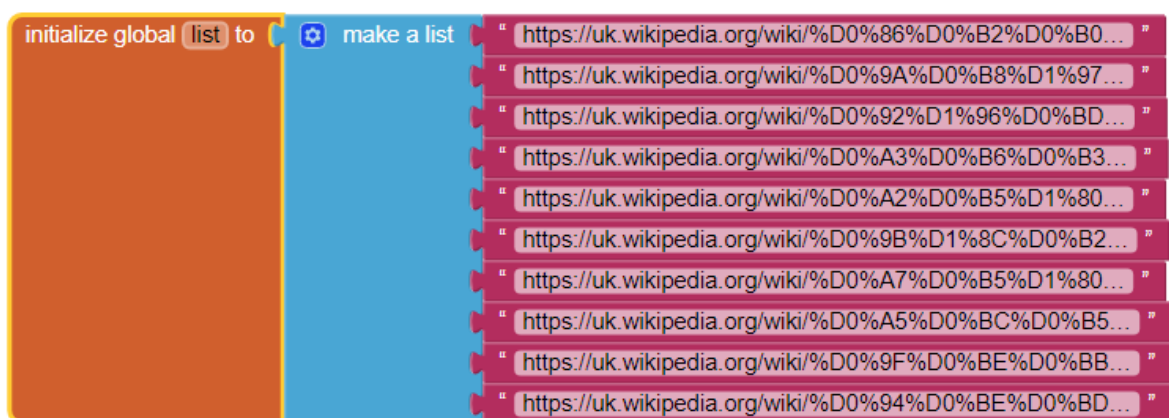


**16** **В залежності від того, яка кнопка була натиснута, твій переглядач має завантажити потрібне посилання.**

**Щоб максимально скоротити програму, скористайся списками.**

**Створи список, елементами якого будуть посилання на обрані сторінки з інформацією.**

***Примітка. Елементи обов'язково розташовуй в тому ж порядку, що й кнопки.***





17

Опиши процедуру ініціалізації екрану.

```
when web.Initialize
do call WebView1.GoToUrl url
```

18

Так само, як і в інших мовах програмування, ти можеш звернутися до будь-якого елемента списку за допомогою його індексу (порядковий номер).

```
select list item list
index
```

Для цього тобі необхідний такий блок , в якому потрібно вказати назву списку та номер елемента.

Склади такий скрипт

```
when web.Initialize
do call WebView1.GoToUrl url
select list item list
index
get global list
get start value
```

19

Склади скрипти для кнопок «Вперед», «Назад» та «Вихід»

```
when back.Click
do call WebView1.GoBack
```

```
when forward.Click
do call WebView1.GoForward
```

```
when exit.Click
do open another screen screenName "Screen1"
```



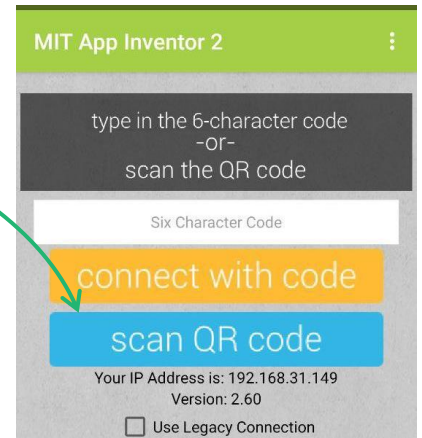
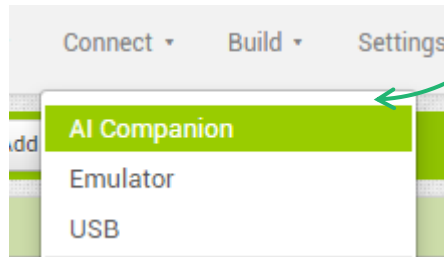
## Випробуй свій проект

20

Відкрий на смартфоні додаток **MIT AI2 Companion** та натисни кнопку **Scan QR code**.

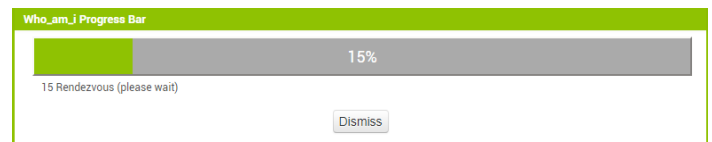
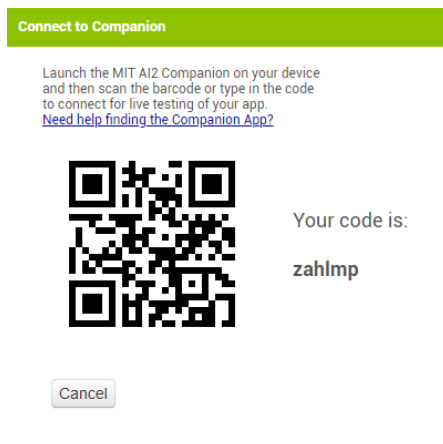
21

Відкрий меню **Connect** в середовищі **App Inventor** та вибери зі списку



22

Відскануй смартфоном QR код, який відкрився та зачекай завантаження твого додатку



23

Випробуй свій додаток.



## Самостійне завдання

**Зроби твій довідник «зламаним».**

**Поломка має полягати в тому, що замість правильної інформації, користувачу відкриватиметься неправильне посиланням.**

**Додай ще одну кнопку, яка буде ламати, або ж налагоджувати роботу програми.**

***Примітка 1. Обов'язково використовуй рандом.***

***Примітка 2. Передбач можливість того, що у зламаному режимі в один момент все ж таки може згенеруватись правильна сторінка.***

Що буде в цьому проєкті?

Ти створиш «Вікторину», з різними типами питань

Навички, які ти отримаєш

Ти поглибиш знання таких елементів, як Check Box, Spinner та Text Box.

## Умовні позначення



**Виконай**

**Виконуй ці ІНСТРУКЦІЇ одну за одною**



**Випробуй**

**Надішли програму на свій телефон за допомогою QR коду**



**Подумай**

**Виконай самостійне завдання**

## Кольорові підказки

**Назва компонента**

**Назва властивості**

**Значення властивості**



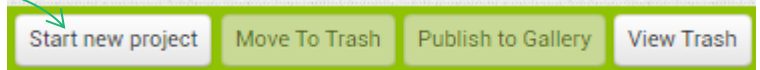
# Частина 1. Розробляємо дизайн

1

Відкрий середовище MIT App Inventor, яке знаходиться за посиланням <http://ai2.appinventor.mit.edu/>

2

Створи новий проєкт



**Примітка. Давай своїм проєктам зрозумілі назви**

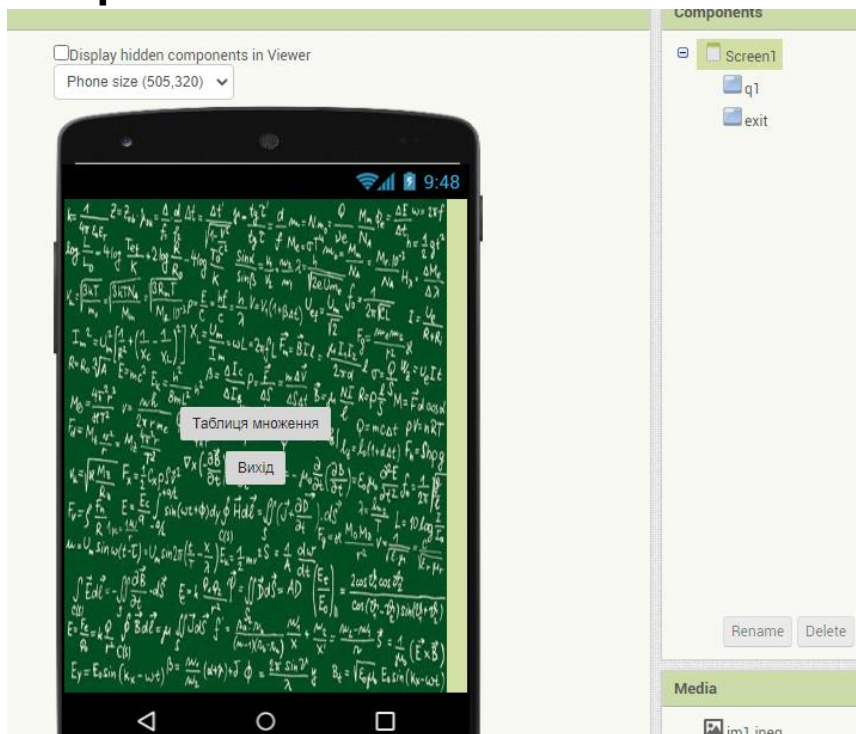
3

Зроби меню.

Додай на екран дві кнопки – щоб розпочати тест та кнопку «Вихід». Розташуй їх на твій розсуд.

За бажанням можна змінити фон екрану.

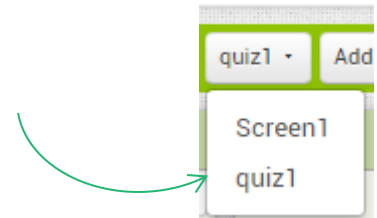
Ось такий екран вийшов в мене.



**Примітка. Як ти бачиш, тематикою моєї вікторини є математика. Ти можеш вибрати іншу – ту, яка подобається саме тобі («Що це за книга?», «Що це за фільм?», «Наскільки добре ти знаєш породи собак?» тощо) .**

5

Створи новий екран **quiz1** та відкрий його



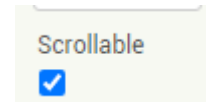
На цьому екрані буде відбуватись опитування гравця.

В тебе буде три типи питань:

- Тест з однією правильною відповіддю (**прапорці**)
- Тест з однією правильною відповіддю (**розкривний список**)
- Розгорнута відповідь

6

Додай можливість прокрутки екрану.



7

Створи тест з прапорцями

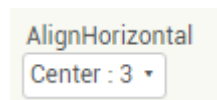


Додай на екран блок **Vertical Arrangement**, в якому будуть міститись всі необхідні елементи для цього завдання. Встанови **ширину** для нього на

повний екран



та центральне розміщення

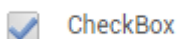


Перетягни в середину цього блоку **Label**



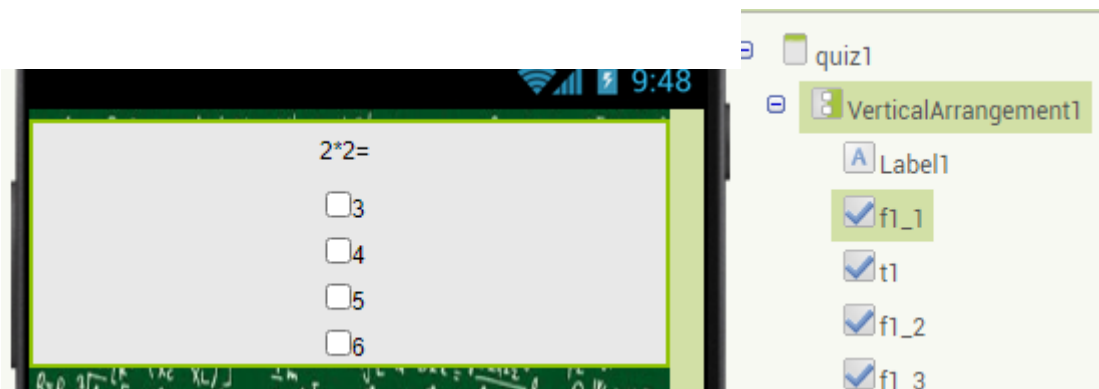
(питання) та чотири

**Check Box**



(варіанти відповідей).

Додай прапорцям такі ж назви, як показано на зображенні.



8

Створи тест з розкривним списком

Додай на екран блок **Vertical Arrangement**, в якому будуть міститись всі необхідні елементи для цього завдання. Встанови **ширину** для нього на



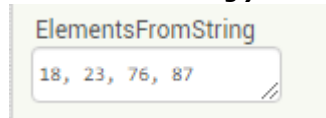
повний екран та центральне розміщення.

Перетягни в середину цього блоку **Label** (питання), розкривний

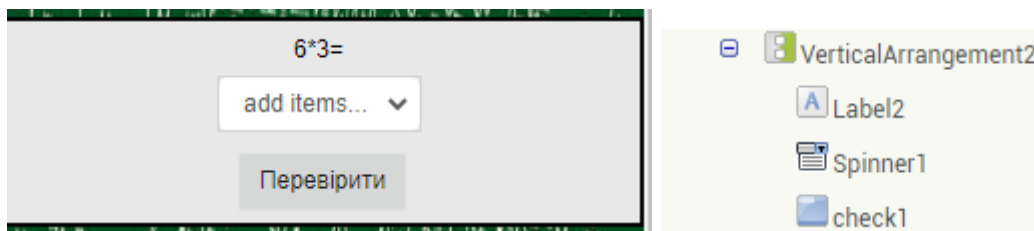
список **Spinner** та кнопку «Перевірити».

Додай варіанти відповідей (елементи списку) для **Spinner** за допомогою

властивості **Elements From String**



**Примітка: Елементи списку розілюються комою**

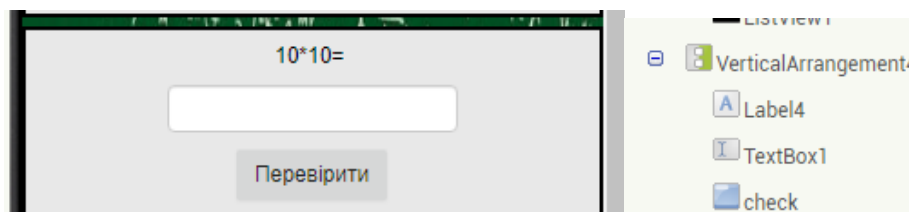


## 9

Створи питання з відкритою відповіддю

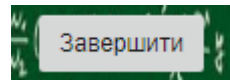
Додай знову на екран блок **Vertical Arrangement** та встанови для нього такі ж властивості, як і в попередніх кроках.

Перетягни в середину цього блоку **Label** (питання), **текстове поле** **TextBox** (відповідь) та кнопку «Перевірити».



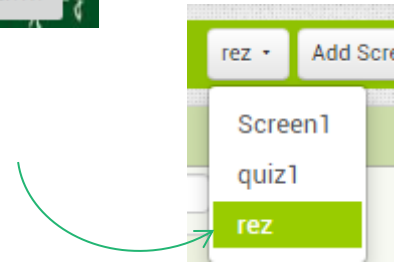
10

Додай на екран кнопку «Завершити»



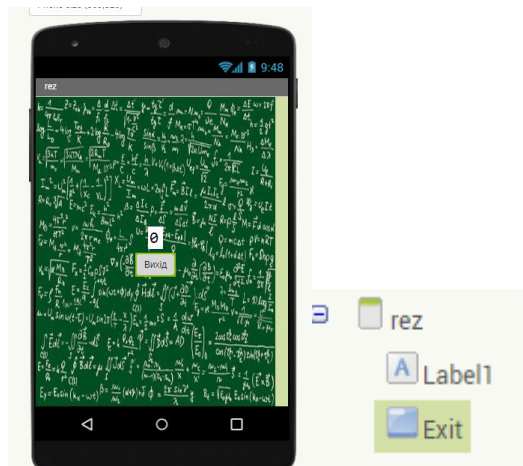
11

Створи новий екран rez та відкрий його.



12

На результуючий екрані тобі потрібно додати тільки Напис (кількість балів) та кнопку «Вихід»





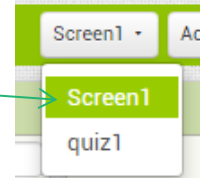


## Частина 2. Програмуємо елементи

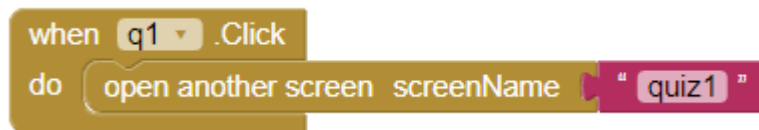
- 13 **Перейди в режим розробника.**



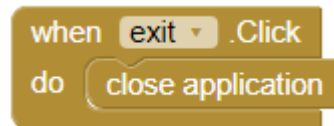
- 14 **Розпочни програмування з початкового екрану.**



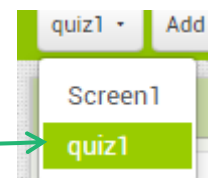
- 15 **Запрограмуй перехід на екран із Вікториною.**



- 16 **Запрограмуй вихід з програми**



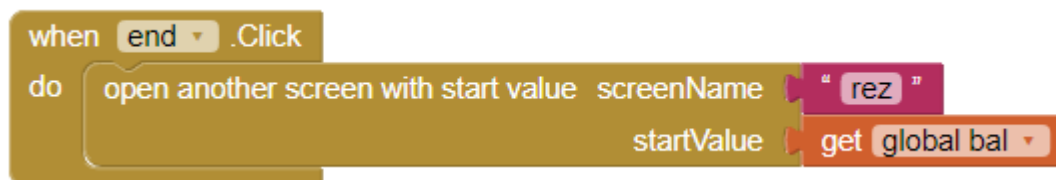
- 17 **Перейди на екран quiz1**



- 18 **Ініціалізуй змінну для обрахунку результату.**



- 19 **Опиши кнопку «Завершити»**



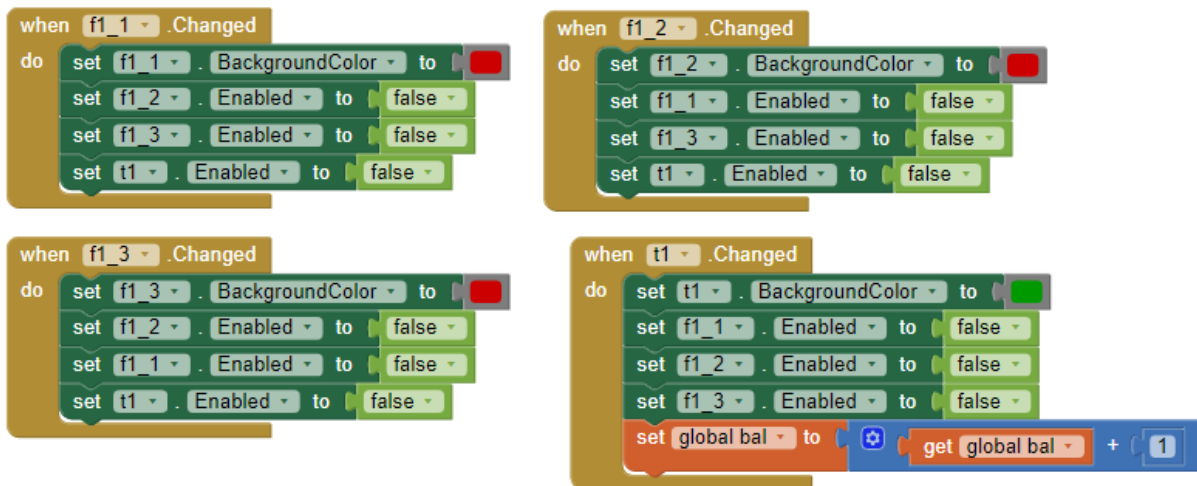
- 20 **Склади програму для питання з прапорцями.**

Для того, щоб прописати тест, тобі знадобиться скласти програму для кожного прапорця окремо.

В тебе може бути два випадки: вибрана відповідь є правильною або хибною.

Щоб користувач бачив, правильно він відповів чи ні, змінюй фон обраного прапорця на зелений або червоний. Також, роби інші прапорці недоступними.

І найголовніше. Якщо обрано вірний прапорець – збільш кількість балів.



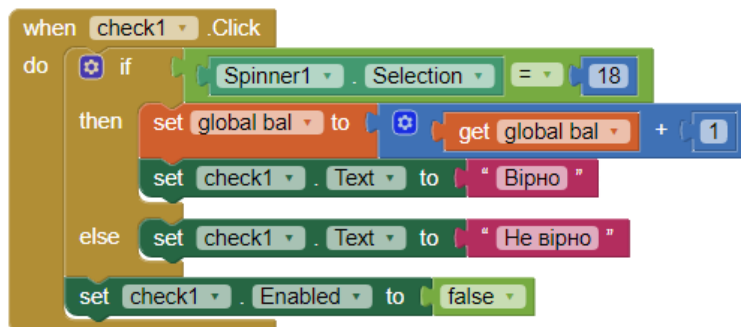
## 21

Склади програму для питання з розкривним списком.

Перевірка правильності відповіді буде здійснюватися після того, як користувач натисне на відповідну кнопку.

Так само як і з прапорцями, може бути два шляхи – сума збільшується або ні, тож тобі потрібно зробити розгалуження.

Щоб користувач бачив, чи правильно він відповів – змінюй текст на кнопці, а щоб забрати в нього можливість відповісти двічі – зроби кнопку недоступною.



## 22

Запрограмуй питання з відкритою відповіддю.

Код для нього буде практично аналогічним до попереднього.

```
when check2 .Click
do
  if TextBox1 . Text = " 100 "
  then
    set global bal to get global bal + 1
    set check2 . Text to " Вірно "
  else
    set check2 . Text to " Не вірно "
  set check2 . Enabled to false
```



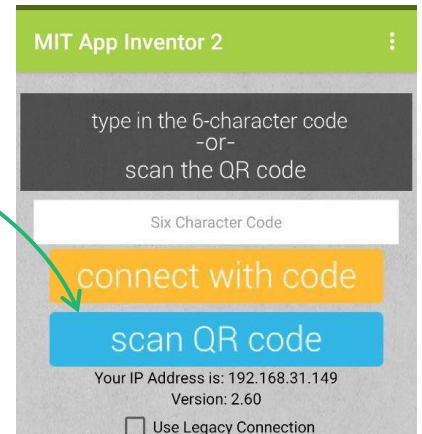
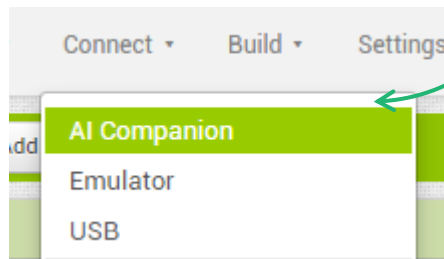
## Випробуй свій проект

23

Відкрий на смартфоні додаток **MIT AI2 Companion** та натисни кнопку **Scan QR code**.

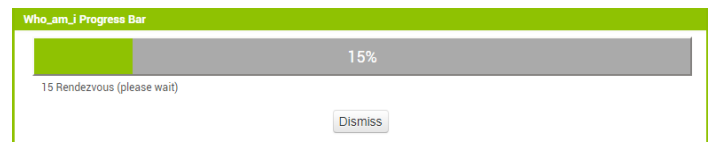
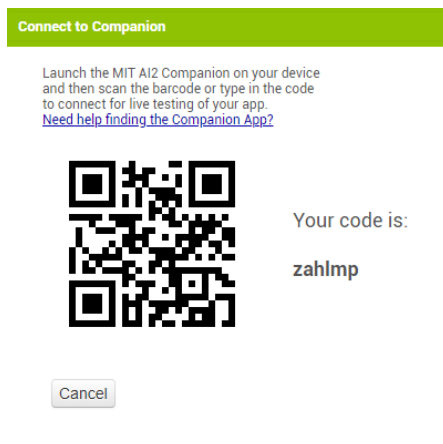
24

Відкрий меню **Connect** в середовищі **App Inventor** та вибери зі списку



25

Відскануй смартфоном QR код, який відкрився та зачекай завантаження твого додатку



26

Випробуй свій додаток.



## Самостійне завдання

- 1. Запрограмуй елементи на кінцевому екрані. Після ініціалізації, текст напису має змінитись на кількість отриманих балів.**
- 2. Додай до Вікторини 10-15 питань (тип питання можеш обрати самостійно)**
- 3. Додай до тесту часове обмеження (таймер). Після того, як час вийде, автоматично здійснюється перехід на кінцеву сторінку.**

Що буде в цьому проєкті?

Створення додатка, за допомогою якого можна переглядати інформацію про Wi-Fi підключення, підключатись до нових мереж

Навички, які ти отримаєш

Ти дізнаєшся, що таке імпортовані компоненти, де їх можна взяти та як використовувати.

## Умовні позначення



**Виконай**

Виконуй ці ІНСТРУКЦІЇ одну за одною



**Випробуй**

Надішли програму на свій телефон за допомогою QR коду



**Подумай**

Виконай самостійне завдання

## Кольорові підказки

**Назва  
компонента**

**Назва  
властивості**

**Значення  
властивості**



## Частина 1. Розробляємо дизайн

1

Завантаж файл **WiFi.aix** в папку проєкту на комп'ютері.

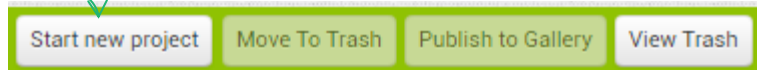
**Примітка.** Цей файл – це розширення до базових можливостей MIT App Inventor. Більше розширень ти можеш знайти на сторінці <https://puravidaapps.com/extensions.php>

2

Відкрий середовище MIT App Inventor, яке знаходиться за посиланням <http://ai2.appinventor.mit.edu/>

3

Створи новий проєкт



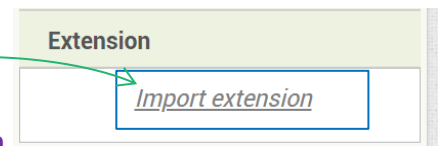
**Примітка.** Давай своїм проєктам зрозумілі назви

4

В MIT App Inventor є можливість завантажувати сторонні компоненти – розширення.

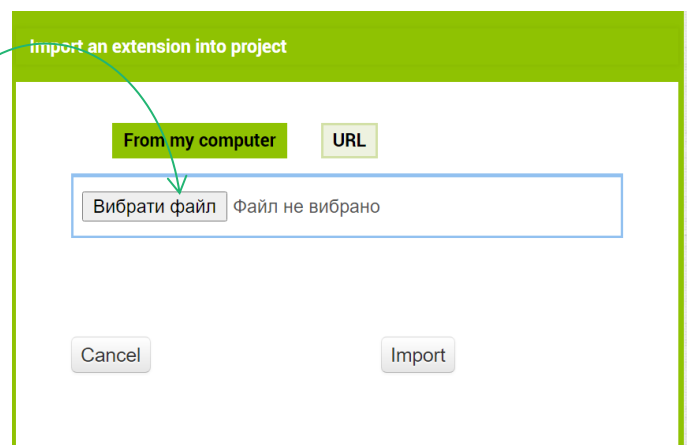
Це можливо зробити в **Palette -> Extension**.

Для цього, потрібно натиснути на **Import extension**



Тепер, натисни на кнопку **Вибрати файл** та обери з твого комп'ютера файл **WiFi.aix**

Натисни на кнопку **Import**



*Import extension*



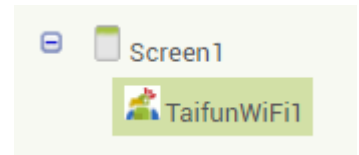
TaifunWiFi



Тепер, у списку в тебе є компонент, в якому вже прописані деякі з методів, які ти будеш сьогодні використовувати.

**5**

Перенеси імпортований компонент на екран. Цей компонент є невидимим, тож на екрані він ніяк не відобразиться, але ти можеш знайти його в списку доданих компонентів.

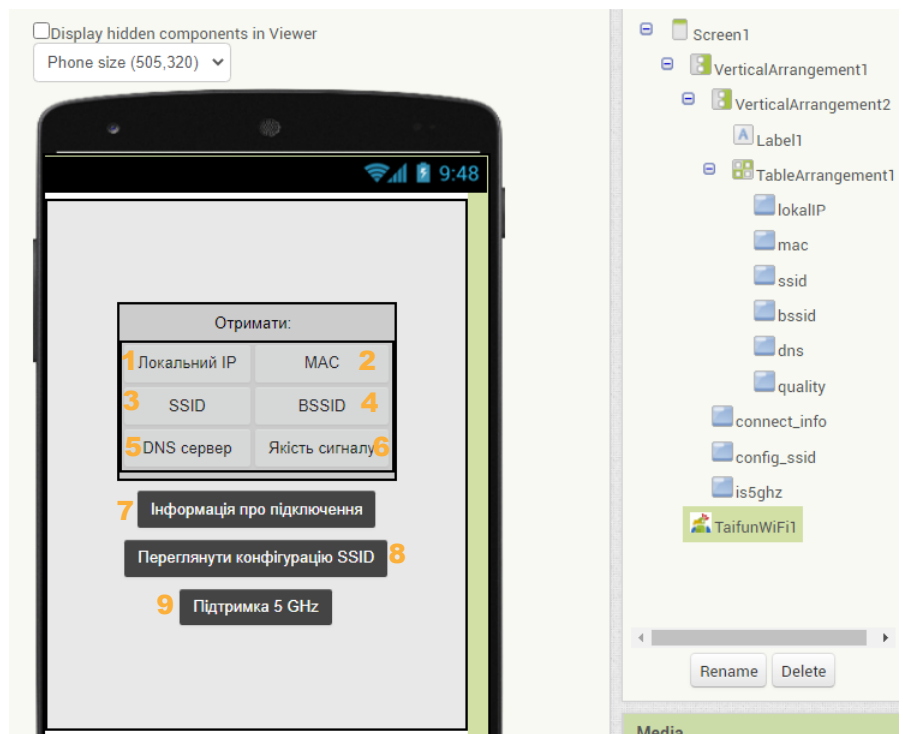
**6**

Зроби заголовок програми невидимим.

**7**

Додай такі кнопки у свою програму:

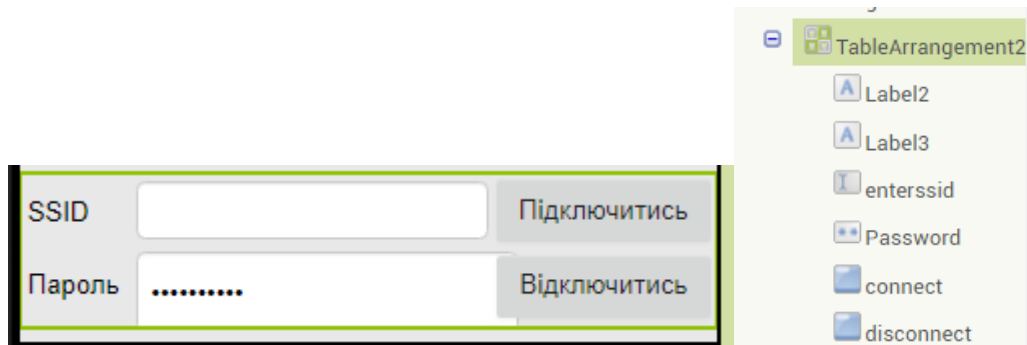
- 1) Отримати локальну IP адресу
- 2) Отримати локальну MAC адресу
- 3) Отримати SSID (назву мережі)
- 4) Отримати BSSID (MAC адреса точки доступу)
- 5) Отримати DNS сервер
- 6) Перевірити якість сигналу
- 7) Переглянути інформацію підключення
- 8) Переглянути конфігурацію (налаштування) точки доступу
- 9) Перевірити, чи підтримує точка доступу діапазон 5 GHz

**8**

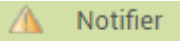
Додай можливість підключитися до мережі.

Для цього, потрібно отримати назву бажаної мережі та пароль (якщо вона захищена).

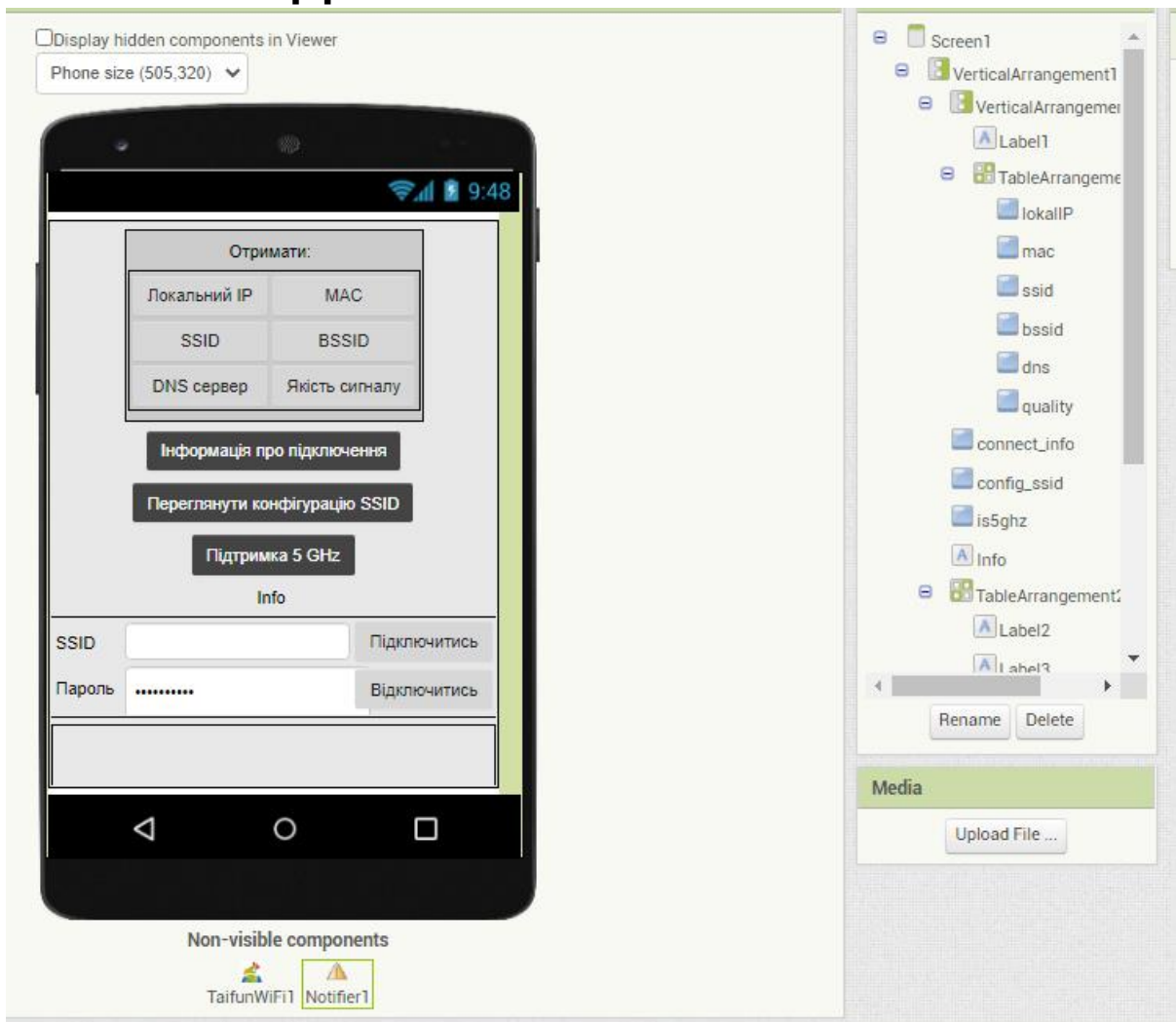




9

Додай на екран компонент **Label** та **Notifier** . Завдяки другому буде здійснюватися відображення попереджень на телефоні. В напис програма буде виводити інформацію, потрібну користувачеві, після натиснення на одну із кнопок.

**Готовий інтерфейс:**





## Частина 2. Програмуємо елементи

10

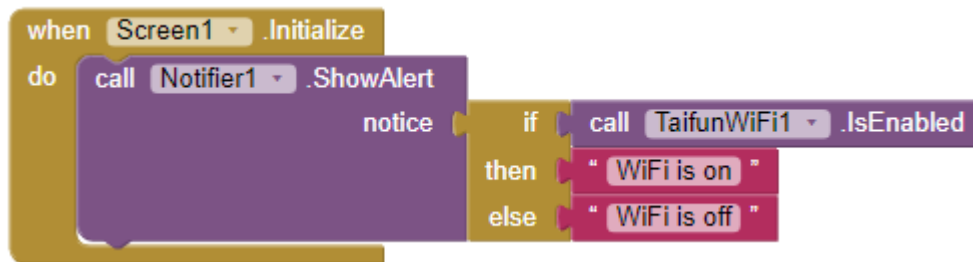
Перейди в режим розробника.

Designer

Blocks

11

Запрограмуй елемент **Notifier**. При відкритті програми користувачу виведеться стан включення **Wi-Fi**.



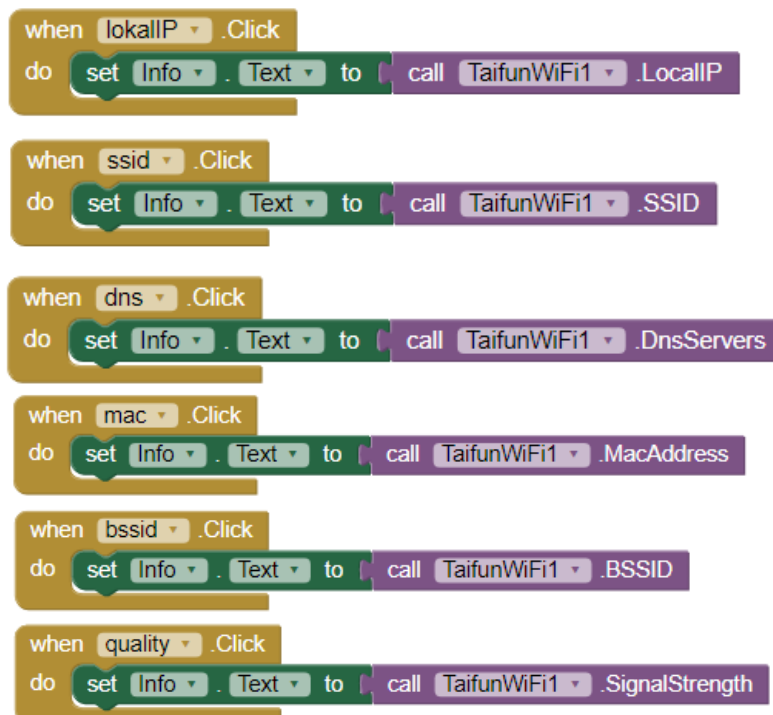
**Примітка:** Блок  **блоків.**

call TaifunWiFi1.IsEnabled

*ти знайдеш у доступних для*

12

Додай блоки, для кнопок із групи «Отримати»



13

Додай дії перегляду інформації про підключення, конфігурації SSID та підтримку 5 GHz

```
when connect_info .Click
do set Info . Text to call TaifunWiFi1 .ConnectionInfo

when config_ssid .Click
do set Info . Text to call TaifunWiFi1 .ConfiguredSSIDs

when is5ghz .Click
do set Info . Text to call TaifunWiFi1 .Is5GHzBandSupported
```

14

Опиши кнопку «Підключитись».

Для цього тобі знадобиться процедура **ConnectSSID**, який є доступним для компонента .

```
when connect .Click
do call enterssid .HideKeyboard
set Info . Text to join " trying to connect to "
enterssid . Text
call TaifunWiFi1 .ConnectSSID
ssid enterssid . Text
password Password . Text
```

15

Додай можливість виведення результату підключення – Підключено або Не підключено.

```
when TaifunWiFi1 .AfterWifiNegotiation
successful
do set Info . Text to join " WiFi Negotiation was "
if get successful
then " successful "
else " NOT successful "
```

16

## Запрограмуй кнопку «Відключитися»

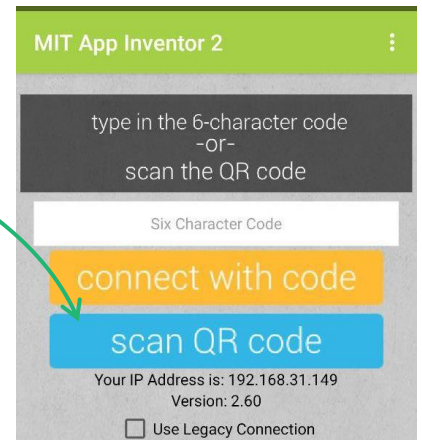
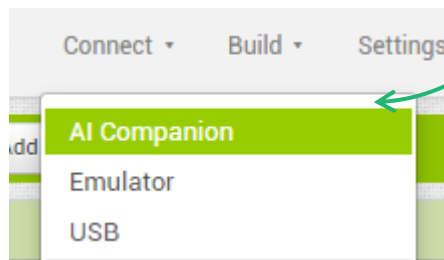
```
when disconnect ▾ .Click  
do call TaifunWiFi1 ▾ .Disconnect
```



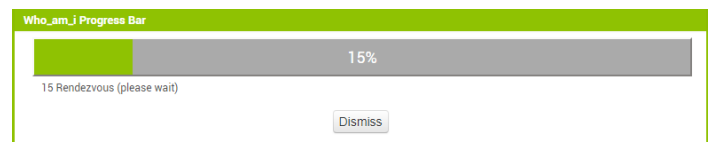
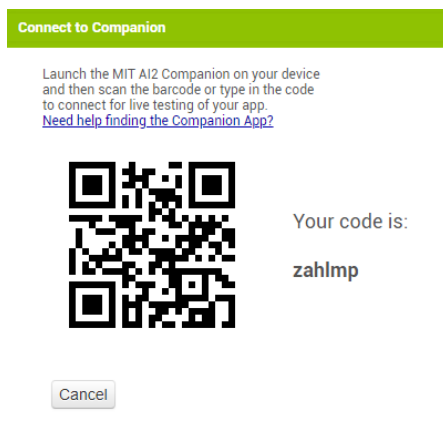
## Випробуй свій проект

**17** Відкрий на смартфоні додаток **MIT AI2 Companion** та натисни кнопку **Scan QR code**.

**19** Відкрий меню **Connect** в середовищі **App Inventor** та вибери зі списку



**19** Відскануй смартфоном QR код, який відкрився та зачекай завантаження твого додатку



**20** Випробуй свій додаток.



## Самостійне завдання

### Спробуй виконати\*

Завантаж зі сторінки <https://puravidaapps.com/extensions.php> будь-яке інше розширення, та створи з цим компонентом додаток.

До кожного з компонентів подана документація, а також – приклад використання в проєктах.

## Використані джерела

1. Інформатика. Навчальна програма для загальноосвітніх навчальних закладів 2–4 класів (оновлена): [Електронний ресурс]. – <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-1-4-klas/5-informatika-2-4-klas.docx>
2. Інформатика для учнів 5-9 класів, які вивчали інформатику у 2-4 класах [Електронний ресурс]. – Режим доступу : <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-5-9-klas/onovlennya-12-2017/8-informatika.docx>
3. НАВЧАЛЬНІ ПРОГРАМИ 10-11 КЛАСІВ. Інформатика Профільний рівень [Електронний ресурс]. – Режим доступу: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/01/10-11-profilniy-riven.docx>
4. Леонова В. І. Волонтерський рух в Україні : методичні рекомендації. / В. І. Леонова. – Одеса : видавець Букаєв Вадим Вікторович, 2019. С.
5. Семеніхіна О. В., Руденко Ю. О. Проблеми навчання програмувати учнів старших класів та шляхи їх подолання Інформаційні технології і засоби навчання : електрон. наук. фах. вид. / Нац. акад. пед. наук України, Ін-т інформ. технол. і засобів навчання, Ін-т модернізації змісту освіти, Ун-т менедж. освіти. – Київ: [б. в.], 2006. С 54-64
6. ІТ-Школа.Че [Електронний ресурс] : [Веб-сайт] — Навіщо вчити дітей програмуванню? — Режим доступу до ресурсу: <http://itschool.ck.ua/navishho-vchyty-ditej-programuvannyu.html>
7. Worldwide Statcounter [Електронний ресурс] : [Веб-сайт]. Mobile Operating System Market Share – Режим доступу до ресурсу: <https://gs.statcounter.com/os-market-share/mobile/worldwide> (дата звернення: 20.04.2021)
8. Futurio [Електронний ресурс] : [Веб-сайт]. – Що таке BYOD? –

Режим доступу до ресурсу: <http://thefuture.news/byod>

9. MIT App Inventor [Електронний ресурс] : [Веб-сайт] — Stories — Режим доступу до ресурсу: <https://appinventor.mit.edu/explore/stories>

10. MIT App Inventor [Електронний ресурс] : [Веб-сайт] — Artificial Intelligence with MIT App Inventor — Режим доступу до ресурсу: <https://appinventor.mit.edu/explore/ai-with-mit-app-inventor>

11. MIT App Inventor [Електронний ресурс]: [Веб-сайт] — Tutorials for MIT App Inventor — Режим доступу до ресурсу: <http://appinventor.mit.edu/explore/ai2/tutorials>

12. MIT App Inventor [Електронний ресурс] : [Веб-сайт] — Teaching with App Inventor — Режим доступу до ресурсу: <http://appinventor.mit.edu/explore/teach>

13. Coursera [Електронний ресурс] : [Веб-сайт] — Developing Android Apps with App Inventor — Режим доступу до ресурсу: <https://www.coursera.org/learn/app-inventor-android>

14. Appinventor.org [Електронний ресурс] : [Веб-сайт] — Course In A Box — Режим доступу до ресурсу: <http://www.appinventor.org/content/CourseInABox/drawAnimate>

15. David Wolber, Hal Abelson, Ellen Spertus, Liz Looney, App Inventor 2 Create Your Own Android Apps, Published by O'Reilly Media, 2015.

16. Антоніна Бучач, Кодуємо для Android Режим доступу до ресурсу: <http://bit.ly/2UFrvPM>

17. Derek Walter, Mark Sherman, Learning MIT App Inventor: A Hands-On Guide to Building Your Own Android Apps, 2014.

18. Karl-Hermann Rollke, Android Apps with App Inventor 2: Easy App Development for Everyone, 2018

19. Типи мобільних додатків URL: <https://agilie.com/ru/blog/mobilnyieprilozhieniia-nativnyie-vieb-i-ghibridnyie> (дата звернення: 17.10.2021)



20. MIT News Office [Електронний ресурс] : [Веб-сайт]. – The MIT roots of Google’s new software ? – Режим доступу до ресурсу: <http://news.mit.edu/2010/android-abelson-0819>

21. MIT App Inventor [Електронний ресурс] : [Веб-сайт]. – Setting Up App Inventor – Режим доступу до ресурсу: <http://appinventor.mit.edu/explore/ai2/setup>

22. MIT App Inventor [Електронний ресурс] : [Веб-сайт]. – App of the Month Winners – Режим доступу до ресурсу: <http://appinventor.mit.edu/explore/app-month-gallery>

23. Цирульник С. М. MIT App Inventor: створення android-додатку лабораторного практикуму без програмування // Відкрите освітнє е-середовище сучасного університету. 2018. Вип. 4. С. 91-95. URL: [http://nbuv.gov.ua/UJRN/oeeemu\\_2018\\_4\\_12](http://nbuv.gov.ua/UJRN/oeeemu_2018_4_12)

24. Appinventor.org [Електронний ресурс] : [Веб-сайт]. – Режим доступу до ресурсу: <http://www.appinventor.org/jbridge>

25. Pura Vida Apps [Електронний ресурс] : [Веб-сайт]. – App Inventor Extensions – Режим доступу до ресурсу: <https://gs.statcounter.com/os-market-share/mobile/worldwide> (дата звернення: 22.05.2021)

26. MIT App Inventor [Електронний ресурс] : [Веб-сайт] — Stories — Режим доступу до ресурсу: <https://thinkable.com/>

27. Meet and Code [Електронний ресурс] : [Веб-сайт]. Про нас – Режим доступу до ресурсу: <https://meet-and-code.org/ua/uk/about>

28. Meet and Code [Електронний ресурс] : [Веб-сайт]. – Онлайн-курс “Дівчата програмують Android додатки” – Режим доступу до ресурсу: <https://meet-and-code.org/be/nl/event-show/4576>

29. Hour of code [Електронний ресурс] : [Веб-сайт]. – Часті питання: <https://hourofcode.com/ua>

30. Євро Освіта [Електронний ресурс] : [Веб-сайт]. – Новітні технології у навчальному процесі – Режим доступу до ресурсу: <http://www.euroosvita.net/prog/print.php/prog/print.php?id=1593>

31. Винничук М. Д., Власій О. О., Гладкий А. І. Можливості використання середовища App Inventor при вивченні інформатики для реалізації міждисциплінарних зв'язків // Дослідження різних напрямів розвитку психології та педагогіки: Збірник наукових робіт учасників міжнародної науково-практичної конференції (15–16 листопада 2019 р., м. Одеса). – Одеса: ГО «Південна фундація педагогіки», 2019. – Ч. 2. – 132 с. URL: <https://cutt.ly/zbSM8q1>.

32. Винничук М. Д., Власій О. О., Гладкий А. І. Розробка вікторини в APP INVENTOR 2 // Комп'ютерні науки, інформаційні технології та системи управління: матеріали Міжнародної науково-технічної конференції здобувачів вищої освіти та молодих вчених, м. Івано-Франківськ, 27–29 листопада 2019 року / наук. ред. Л.Б. Петришин. – Електрон. дані. – Івано-Франківськ : Прикарпатський національний університет імені Василя Стефаника, 2019 – URL: <https://cutt.ly/bcRRhwY>.